IDSProject

2023-12-01

## R Markdown

#EDA

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

##   
## Attaching package: 'arrow'

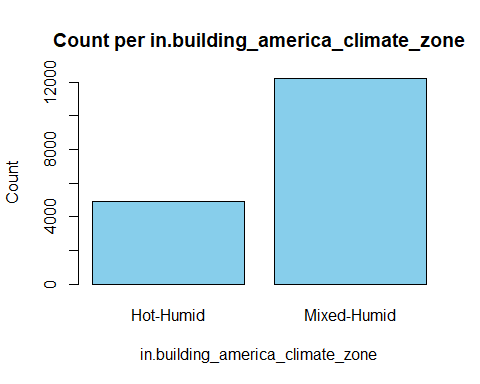
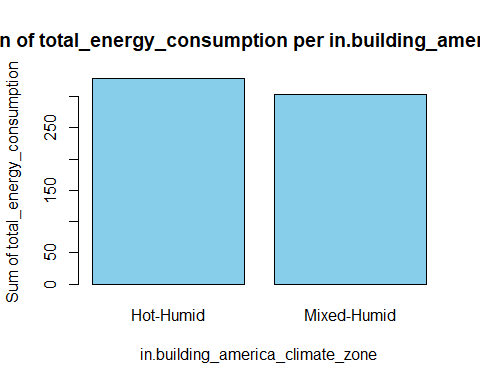
## The following object is masked from 'package:utils':  
##   
## timestamp

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.2 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.3 ✔ 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 become errors

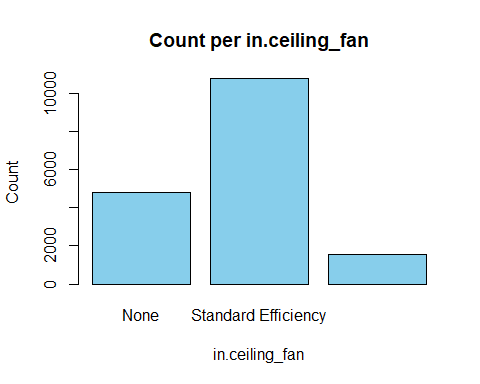
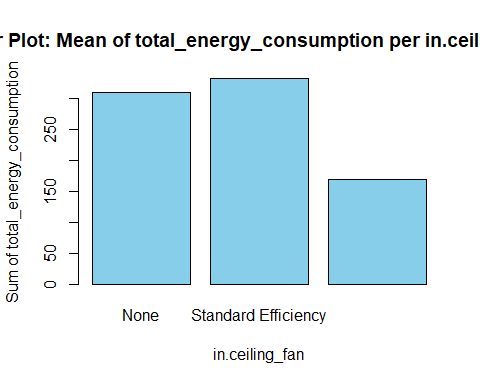
data <- read\_csv("C:/Users/Himanshu/OneDrive/Desktop/Syracuse Sem 1/IST-687 Intro to DS Lab/Final Project/static\_house\_energy\_weather\_df\_17130\_rows\_with\_time\_of\_day.csv", show\_col\_types = FALSE)  
#summary(data)  
View(data)  
  
cols\_req = c('bldg\_id', 'in.sqft', 'in.bedrooms', 'in.building\_america\_climate\_zone', 'in.ceiling\_fan','in.census\_region', 'in.city', 'in.clothes\_dryer', 'in.clothes\_washer', 'in.clothes\_washer\_presence', 'in.cooking\_range', 'in.cooling\_setpoint', 'in.cooling\_setpoint\_has\_offset', 'in.cooling\_setpoint\_offset\_magnitude', 'in.cooling\_setpoint\_offset\_period', 'in.dishwasher', 'in.ducts', 'in.emissions\_fuel\_oil\_values', 'in.emissions\_natural\_gas\_values', 'in.emissions\_propane\_values', 'in.emissions\_wood\_values', 'in.federal\_poverty\_level', 'in.geometry\_attic\_type', 'in.geometry\_floor\_area', 'in.geometry\_floor\_area\_bin', 'in.geometry\_foundation\_type', 'in.geometry\_garage', 'in.geometry\_stories', 'in.geometry\_wall\_exterior\_finish', 'in.geometry\_wall\_type', 'in.has\_pv', 'in.heating\_fuel', 'in.heating\_setpoint', 'in.heating\_setpoint\_has\_offset', 'in.heating\_setpoint\_offset\_magnitude', 'in.heating\_setpoint\_offset\_period', 'in.hot\_water\_fixtures', 'in.hvac\_cooling\_efficiency', 'in.hvac\_cooling\_partial\_space\_conditioning', 'in.hvac\_cooling\_type', 'in.hvac\_has\_ducts', 'in.hvac\_has\_zonal\_electric\_heating', 'in.hvac\_heating\_efficiency', 'in.hvac\_heating\_type', 'in.hvac\_heating\_type\_and\_fuel', 'in.income', 'in.infiltration', 'in.insulation\_ceiling', 'in.insulation\_floor', 'in.insulation\_foundation\_wall', 'in.insulation\_rim\_joist', 'in.insulation\_roof', 'in.insulation\_slab', 'in.insulation\_wall', 'in.interior\_shading', 'in.misc\_extra\_refrigerator', 'in.misc\_freezer', 'in.misc\_gas\_fireplace', 'in.misc\_gas\_grill', 'in.misc\_gas\_lighting', 'in.misc\_hot\_tub\_spa', 'in.misc\_pool', 'in.misc\_pool\_heater', 'in.misc\_pool\_pump', 'in.misc\_well\_pump', 'in.natural\_ventilation', 'in.neighbors', 'in.occupants', 'in.orientation', 'in.plug\_load\_diversity', 'in.pv\_orientation', 'in.pv\_system\_size', 'in.refrigerator', 'in.roof\_material', 'in.tenure', 'in.units\_represented', 'in.usage\_level', 'in.vacancy\_status', 'in.vintage', 'in.vintage\_acs', 'in.water\_heater\_efficiency', 'in.water\_heater\_fuel', 'in.window\_areas', 'in.windows', 'upgrade.insulation\_roof', 'upgrade.water\_heater\_efficiency', 'upgrade.hvac\_cooling\_efficiency', 'upgrade.infiltration\_reduction', 'upgrade.geometry\_foundation\_type', 'upgrade.clothes\_dryer', 'upgrade.insulation\_ceiling', 'upgrade.ducts', 'upgrade.hvac\_heating\_type', 'upgrade.insulation\_wall', 'upgrade.insulation\_foundation\_wall', 'upgrade.hvac\_heating\_efficiency', 'upgrade.cooking\_range', 'time\_of\_day', 'total\_energy\_consumption', 'dry\_bulb\_temperature\_[°c]', 'relative\_humidity\_[%]', 'wind\_speed\_[m/s]', 'wind\_direction\_[deg]', 'global\_horizontal\_radiation\_[w/m2]', 'direct\_normal\_radiation\_[w/m2]', 'diffuse\_horizontal\_radiation\_[w/m2]')  
  
filtered\_data <- select(data, all\_of(cols\_req))  
  
#summary(filtered\_data)  
  
columns\_to\_remove <- c("in.units\_represented")  
filtered\_data <- filtered\_data[, !names(filtered\_data) %in% columns\_to\_remove]

# Assuming filtered\_data is your dataframe  
  
# Select string columns  
string\_columns <- names(filtered\_data)[sapply(filtered\_data, is.character)]  
num\_columns <- names(filtered\_data)[sapply(filtered\_data, is.numeric)]  
  
# Loop through each string column  
for (col in string\_columns) {  
 p\_c <- (which(string\_columns == col)) / length(string\_columns) \* 100  
 print(paste(col, p\_c, '%'))  
   
 # Plotting mean of total\_energy\_consumption per group  
 aggregate(filtered\_data$total\_energy\_consumption ~ filtered\_data[[col]], data = filtered\_data, FUN = mean) -> temp  
 barplot(temp[, 2], names.arg = temp[, 1], main = paste('Bar Plot: Mean of total\_energy\_consumption per', col), xlab = col, ylab = 'Sum of total\_energy\_consumption', col = 'skyblue')  
   
 # Plotting count per group  
 table(filtered\_data[[col]]) -> count\_data  
 barplot(count\_data, main = paste('Count per', col), xlab = col, ylab = 'Count', col = 'skyblue')  
}

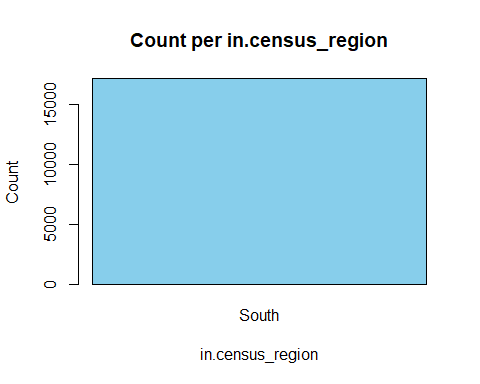
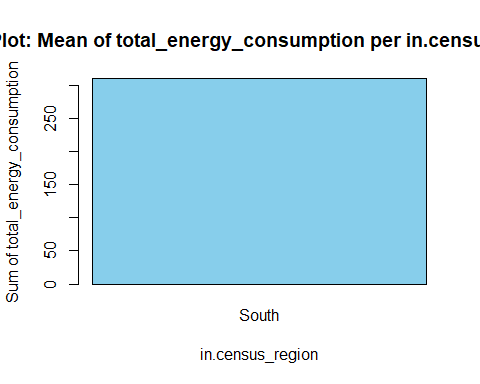
## [1] "in.building\_america\_climate\_zone 1.0989010989011 %"



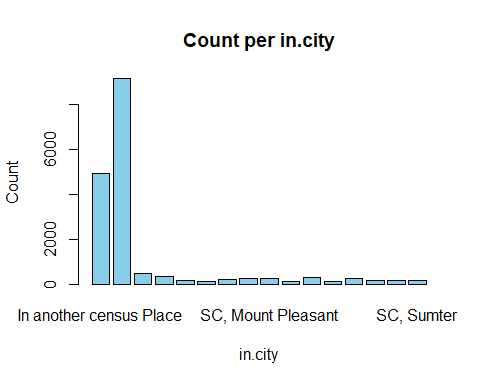
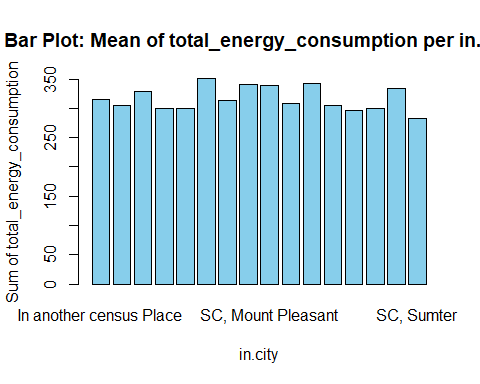
## [1] "in.ceiling\_fan 2.1978021978022 %"



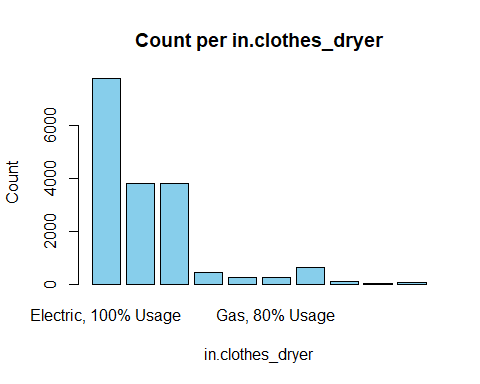
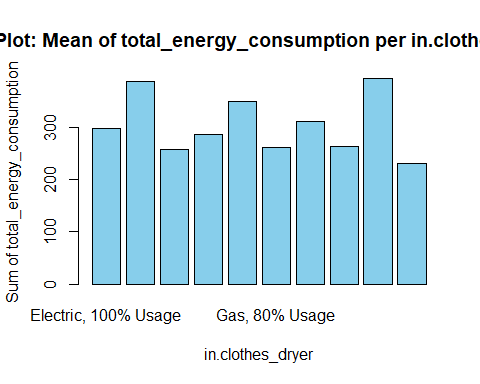
## [1] "in.census\_region 3.2967032967033 %"



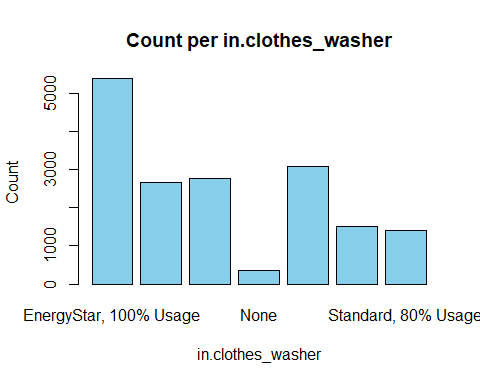
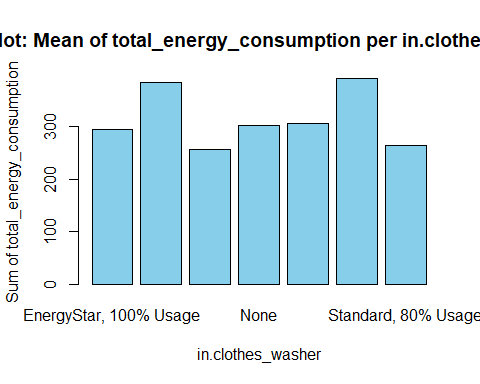
## [1] "in.city 4.3956043956044 %"



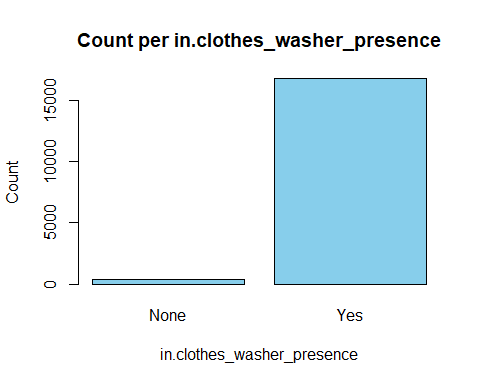
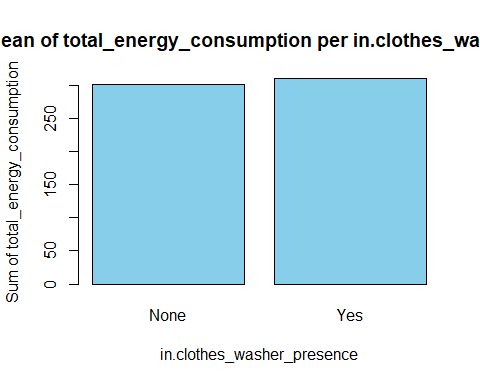
## [1] "in.clothes\_dryer 5.49450549450549 %"



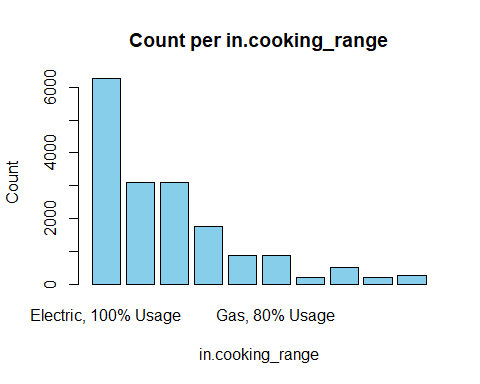
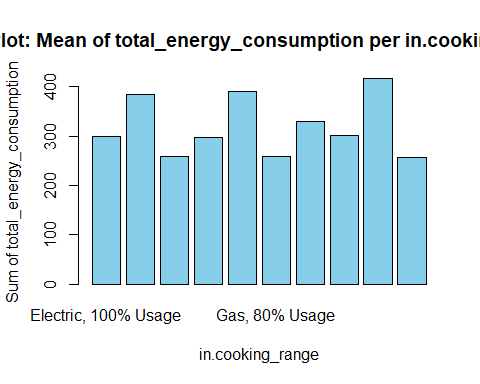
## [1] "in.clothes\_washer 6.59340659340659 %"



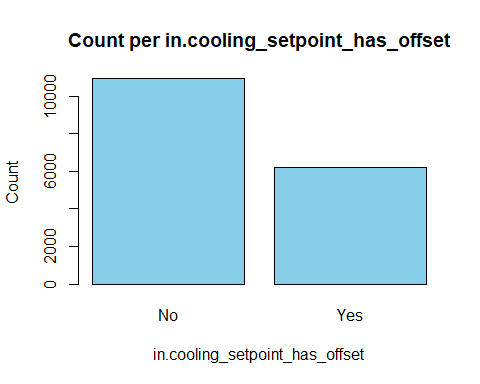
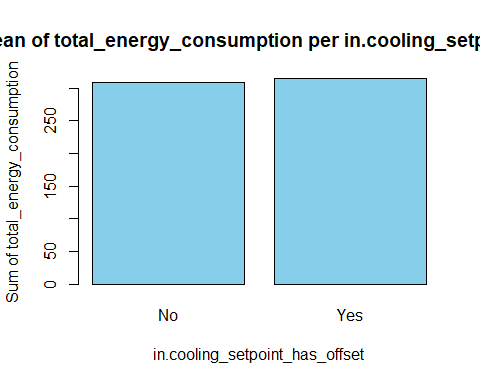
## [1] "in.clothes\_washer\_presence 7.69230769230769 %"



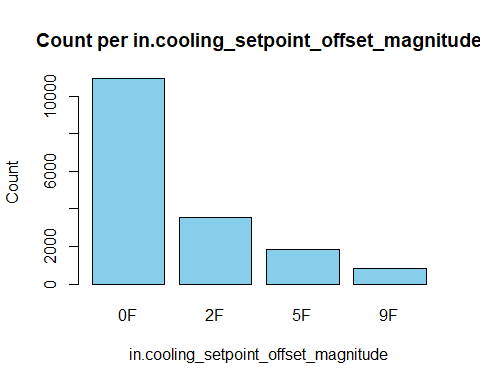
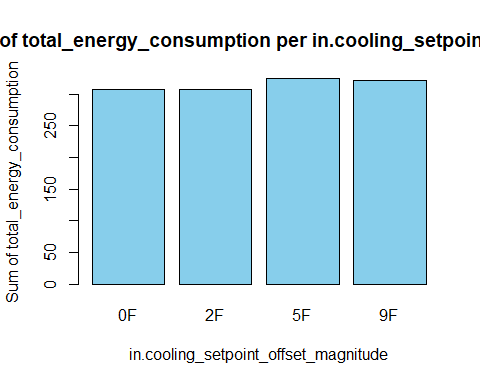
## [1] "in.cooking\_range 8.79120879120879 %"



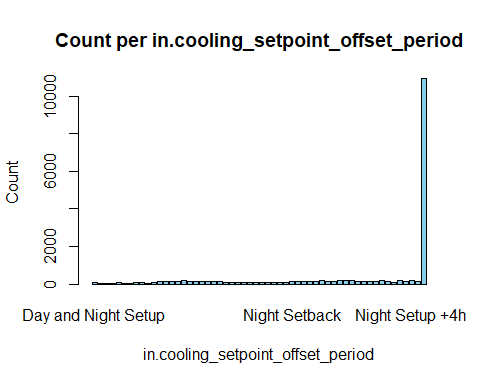
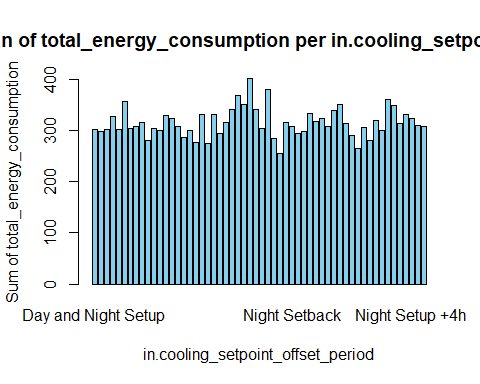
## [1] "in.cooling\_setpoint\_has\_offset 9.89010989010989 %"



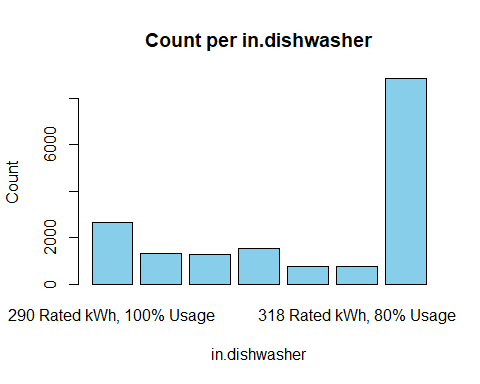
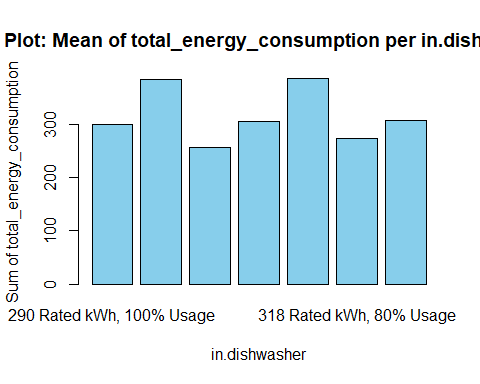
## [1] "in.cooling\_setpoint\_offset\_magnitude 10.989010989011 %"



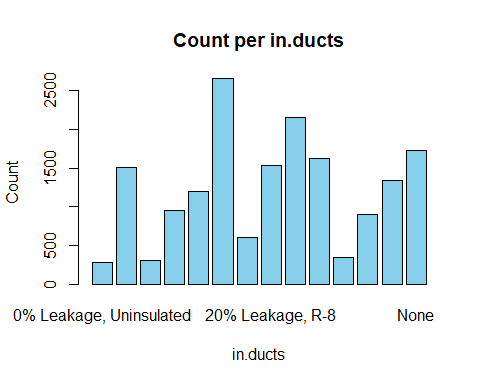
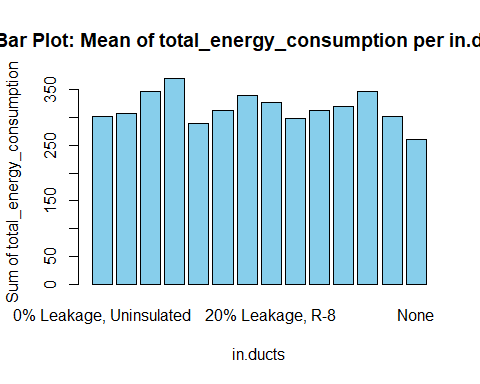
## [1] "in.cooling\_setpoint\_offset\_period 12.0879120879121 %"



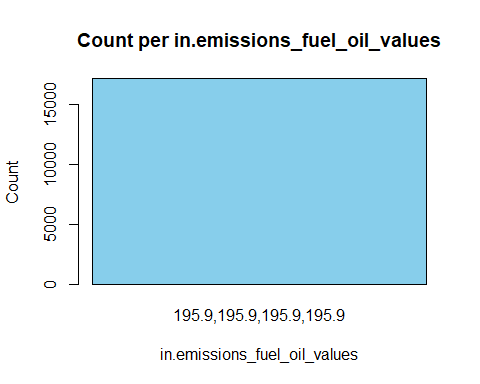
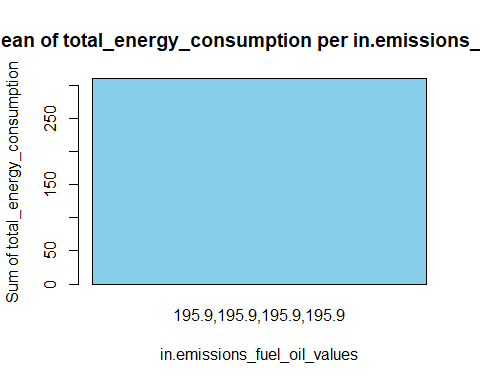
## [1] "in.dishwasher 13.1868131868132 %"



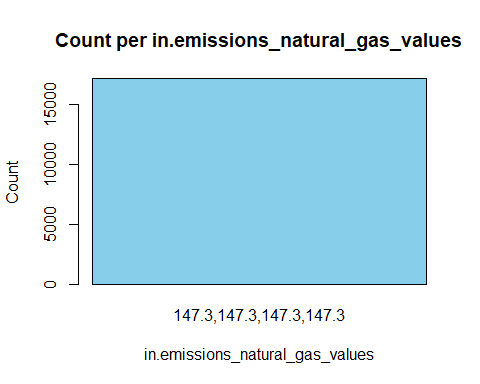
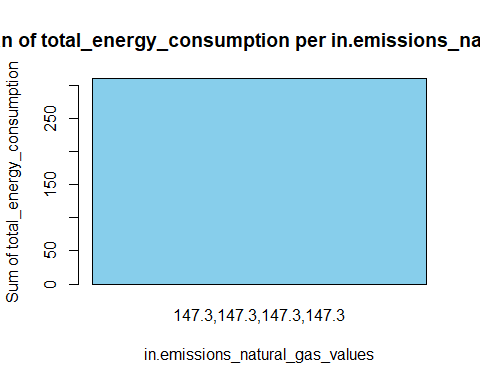
## [1] "in.ducts 14.2857142857143 %"



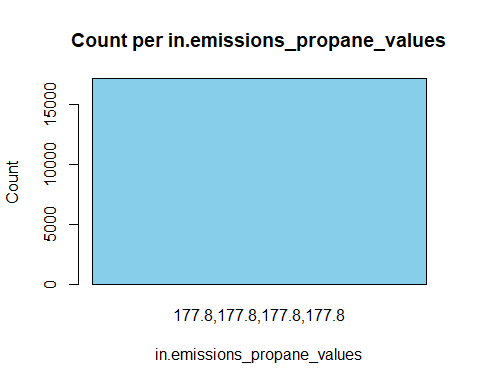
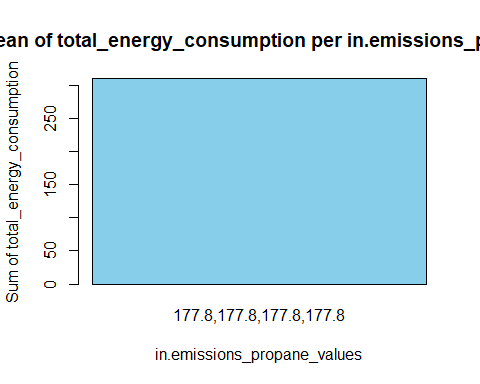
## [1] "in.emissions\_fuel\_oil\_values 15.3846153846154 %"



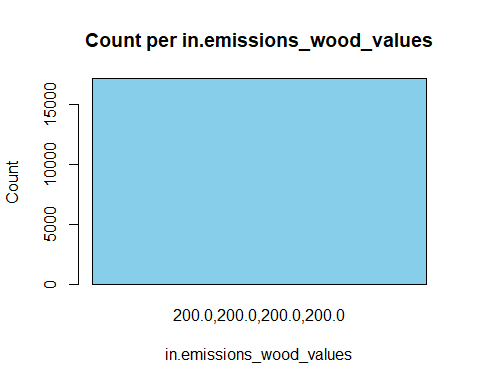
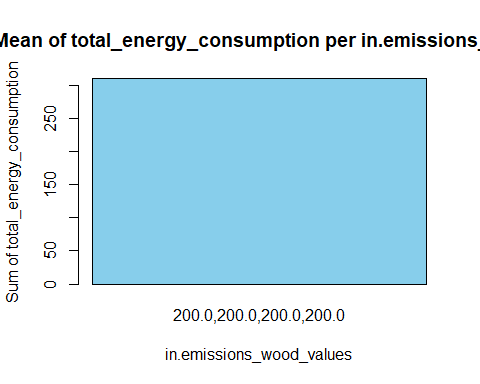
## [1] "in.emissions\_natural\_gas\_values 16.4835164835165 %"



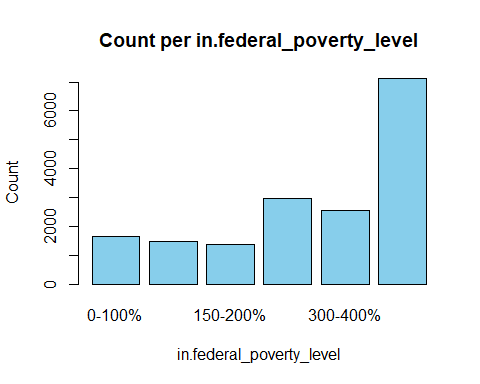
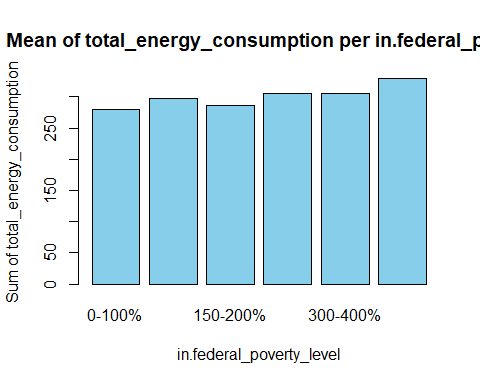
## [1] "in.emissions\_propane\_values 17.5824175824176 %"



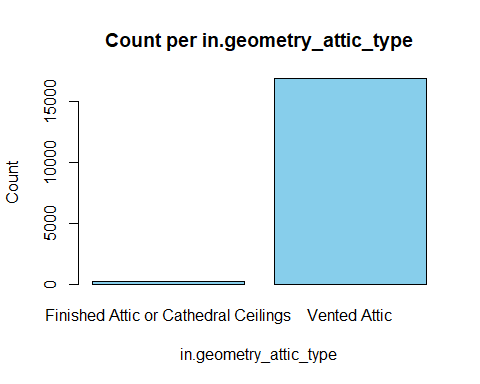
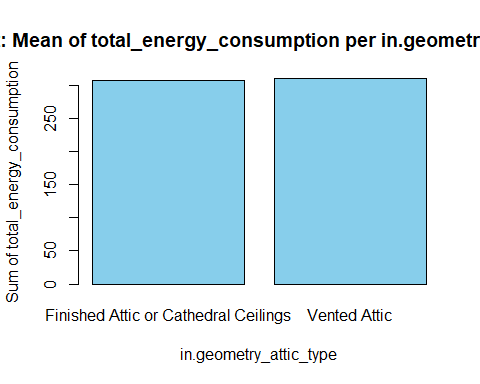
## [1] "in.emissions\_wood\_values 18.6813186813187 %"



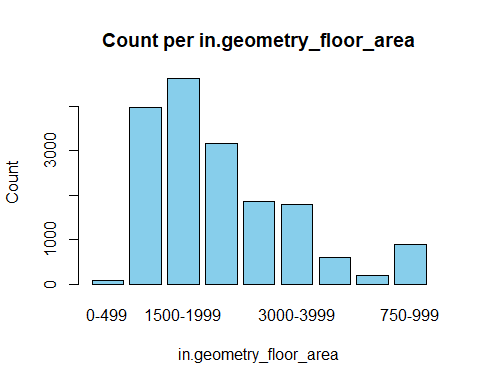
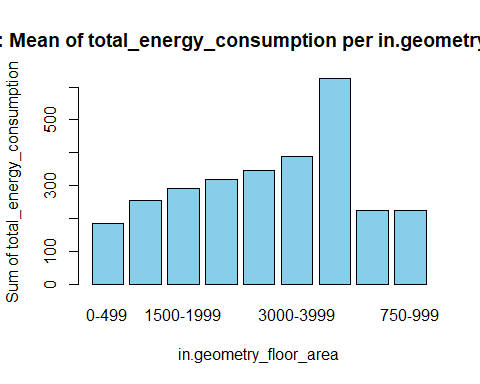
## [1] "in.federal\_poverty\_level 19.7802197802198 %"



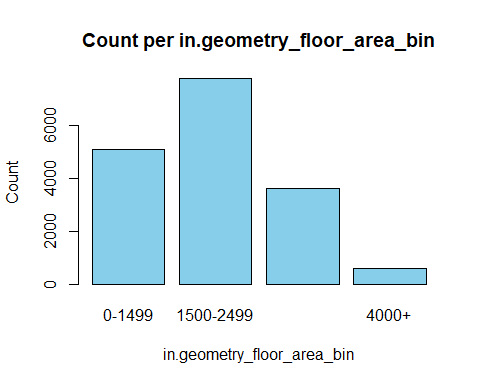
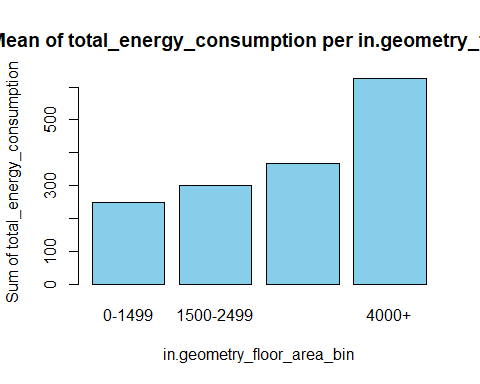
## [1] "in.geometry\_attic\_type 20.8791208791209 %"



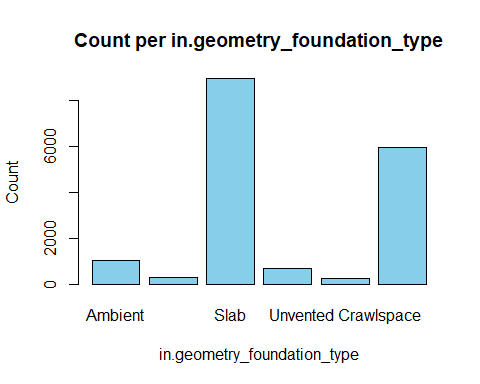
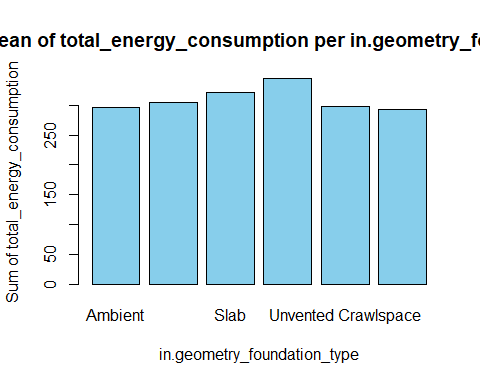
## [1] "in.geometry\_floor\_area 21.978021978022 %"



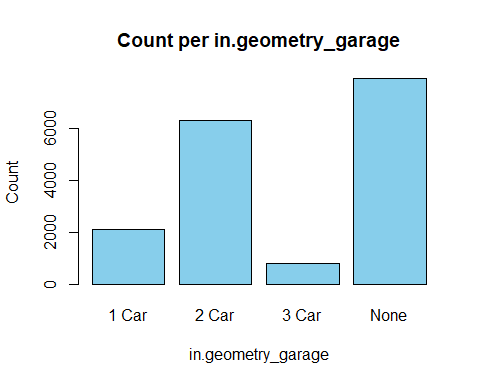
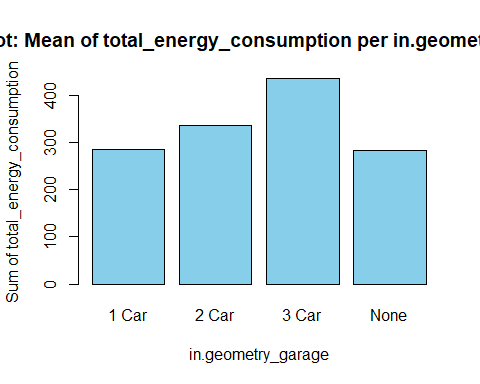
## [1] "in.geometry\_floor\_area\_bin 23.0769230769231 %"



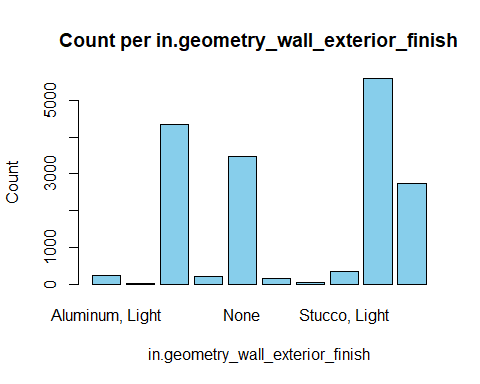
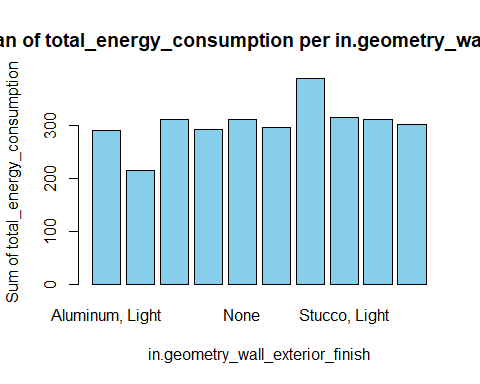
## [1] "in.geometry\_foundation\_type 24.1758241758242 %"



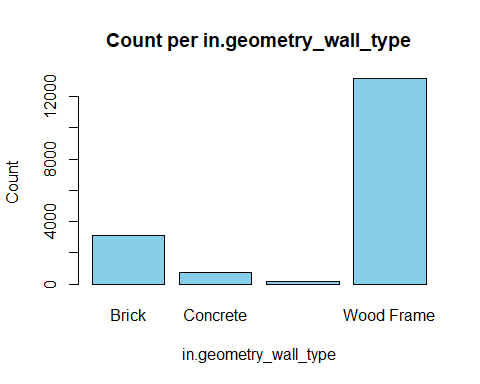
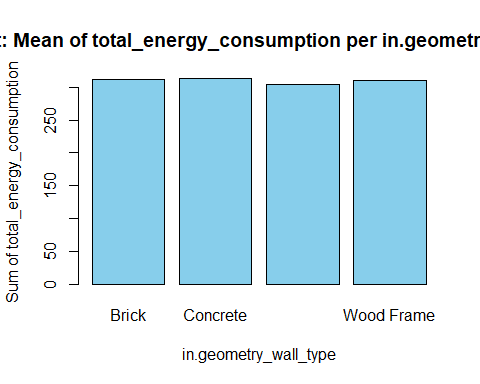
## [1] "in.geometry\_garage 25.2747252747253 %"



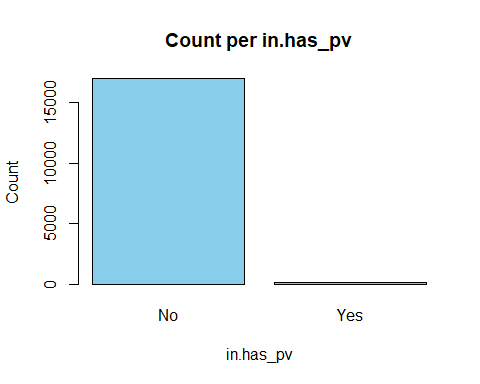
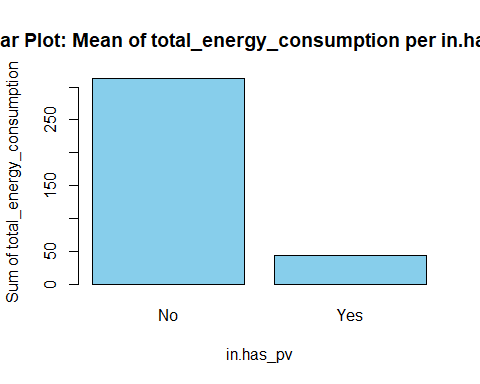
## [1] "in.geometry\_wall\_exterior\_finish 26.3736263736264 %"



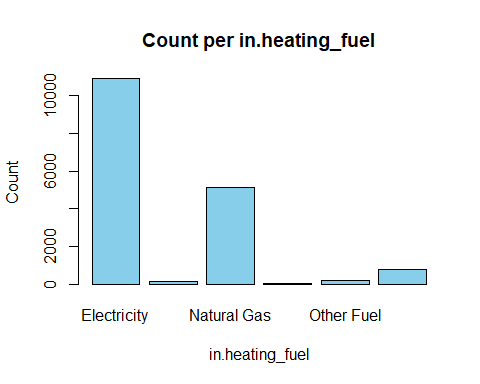
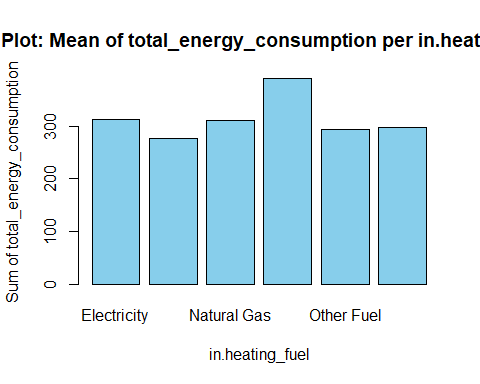
## [1] "in.geometry\_wall\_type 27.4725274725275 %"



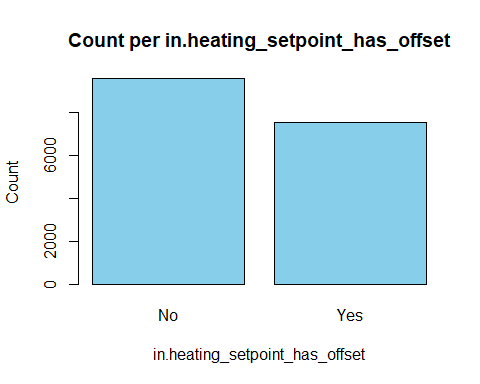
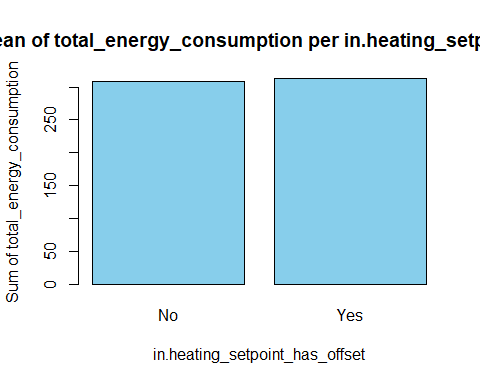
## [1] "in.has\_pv 28.5714285714286 %"



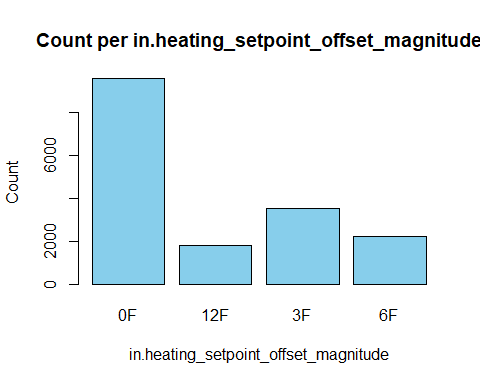
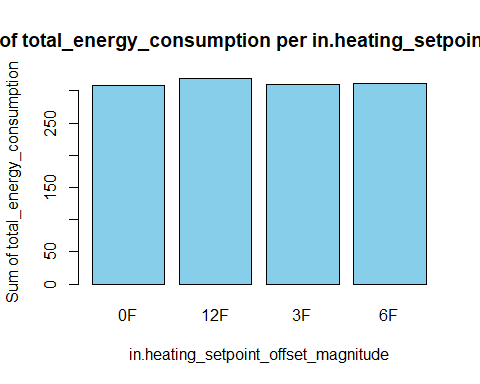
## [1] "in.heating\_fuel 29.6703296703297 %"



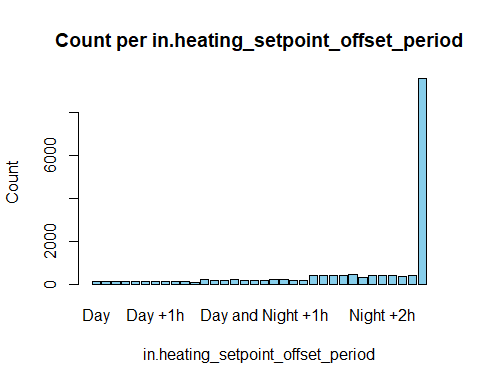
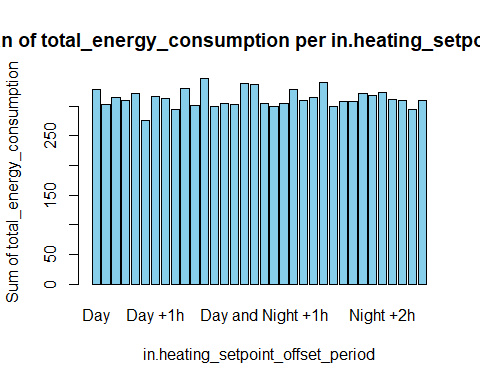
## [1] "in.heating\_setpoint\_has\_offset 30.7692307692308 %"



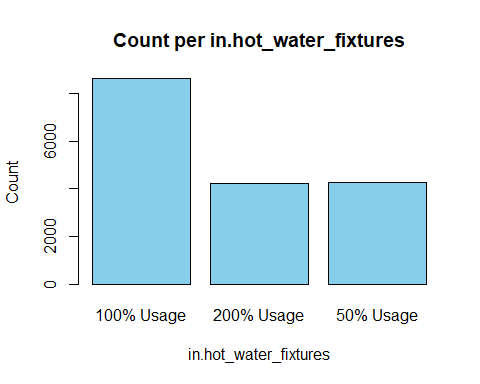
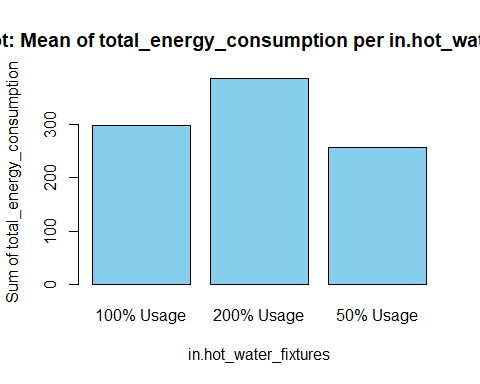
## [1] "in.heating\_setpoint\_offset\_magnitude 31.8681318681319 %"



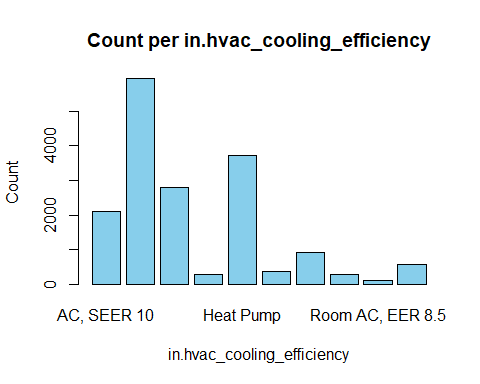
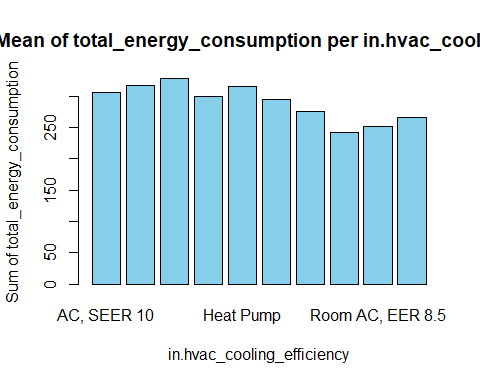
## [1] "in.heating\_setpoint\_offset\_period 32.967032967033 %"



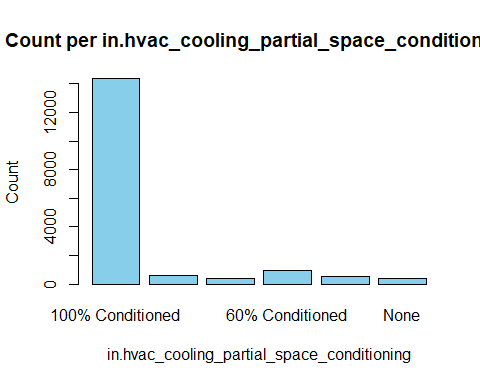
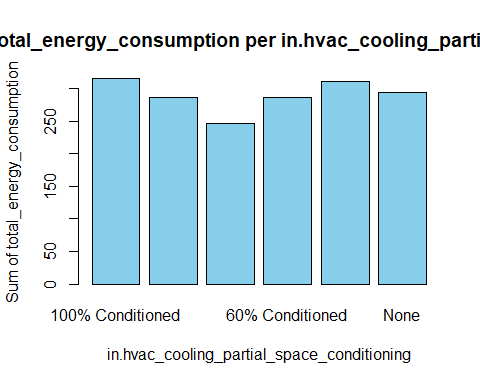
## [1] "in.hot\_water\_fixtures 34.0659340659341 %"



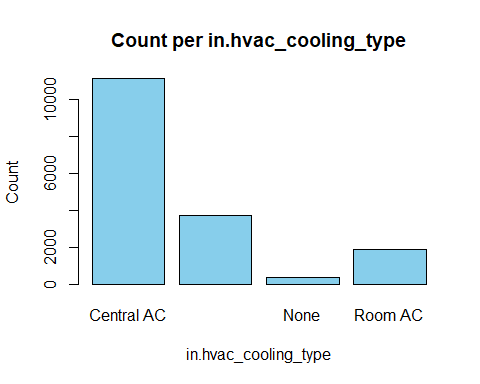
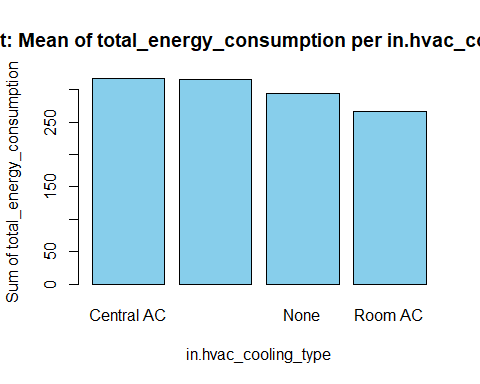
## [1] "in.hvac\_cooling\_efficiency 35.1648351648352 %"



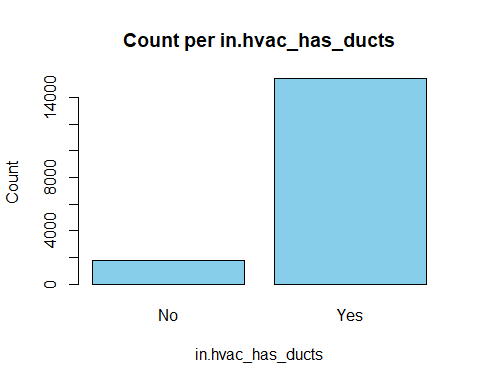
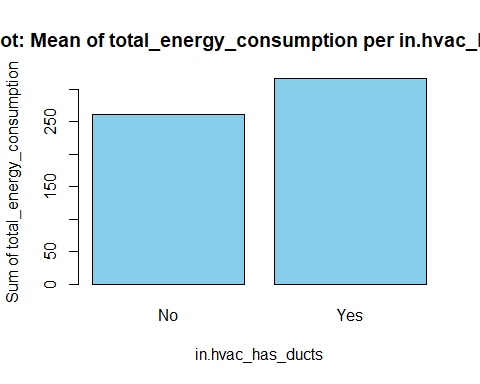
## [1] "in.hvac\_cooling\_partial\_space\_conditioning 36.2637362637363 %"



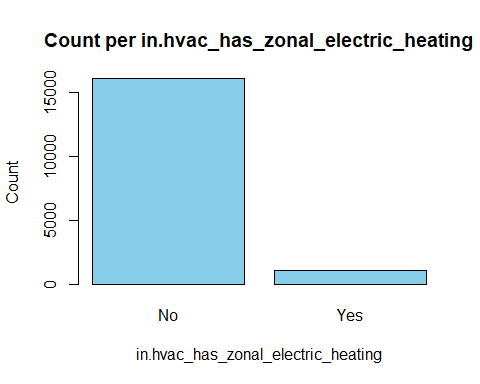
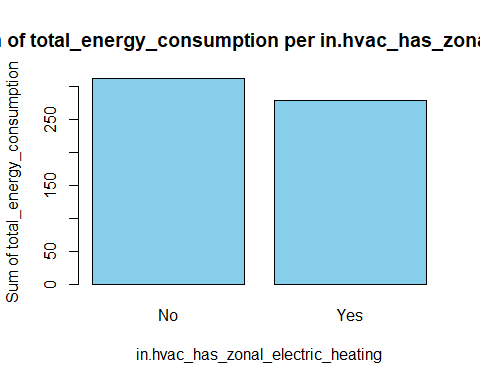
## [1] "in.hvac\_cooling\_type 37.3626373626374 %"



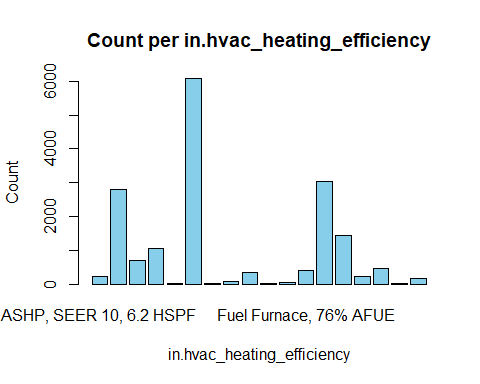
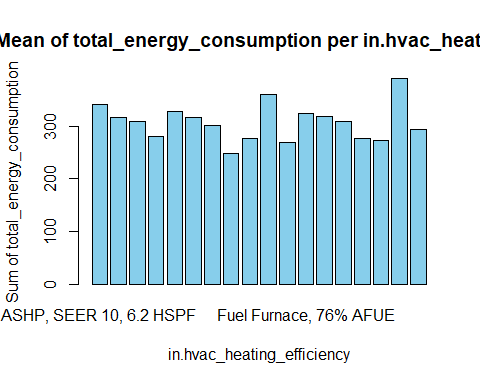
## [1] "in.hvac\_has\_ducts 38.4615384615385 %"



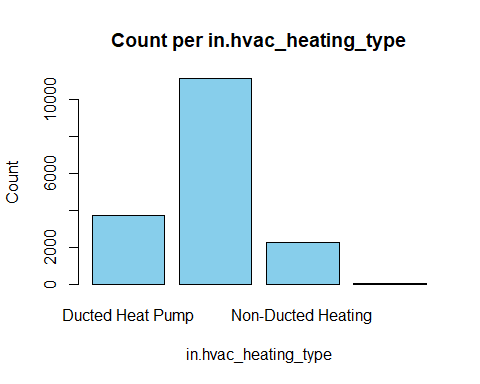
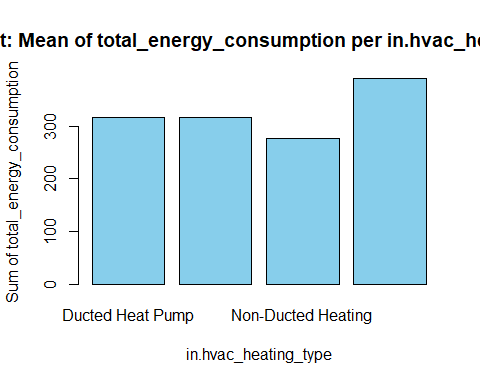
## [1] "in.hvac\_has\_zonal\_electric\_heating 39.5604395604396 %"



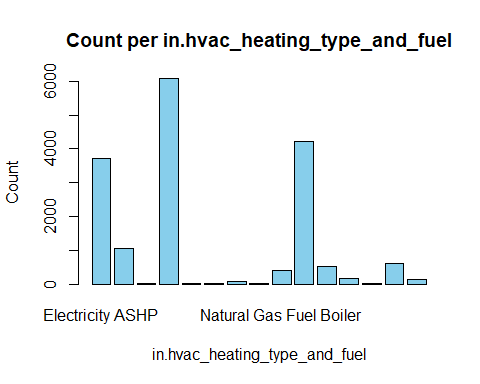
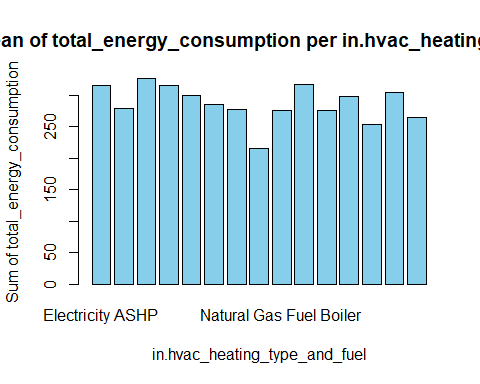
## [1] "in.hvac\_heating\_efficiency 40.6593406593407 %"



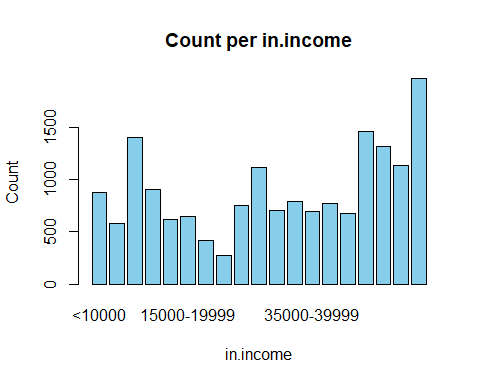
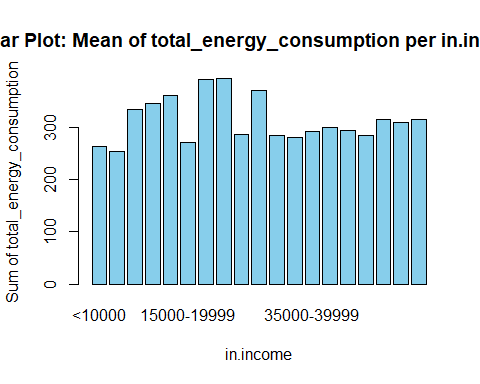
## [1] "in.hvac\_heating\_type 41.7582417582418 %"



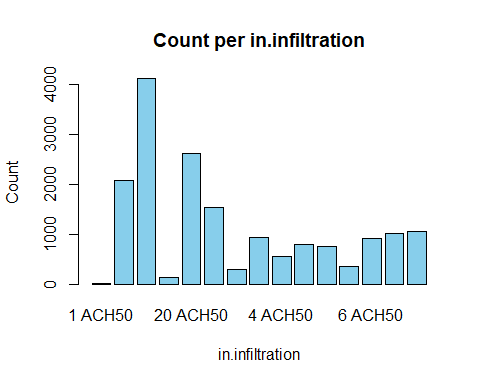
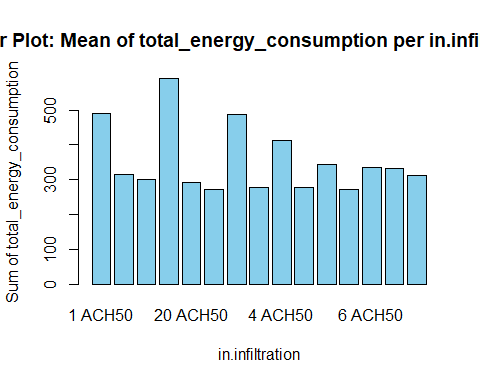
## [1] "in.hvac\_heating\_type\_and\_fuel 42.8571428571429 %"



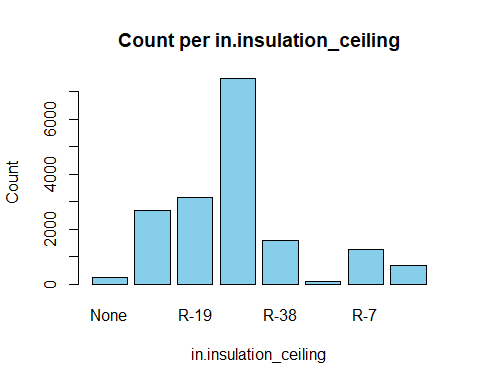
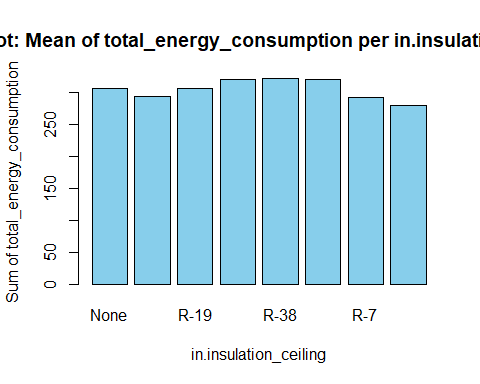
## [1] "in.income 43.956043956044 %"



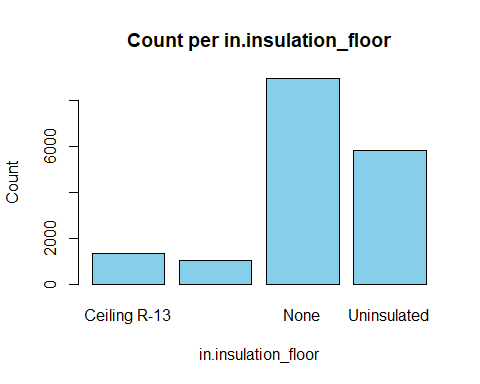
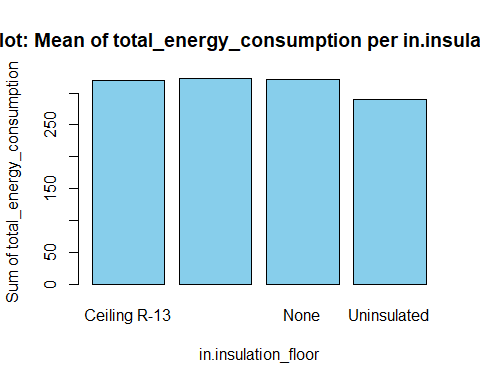
## [1] "in.infiltration 45.0549450549451 %"



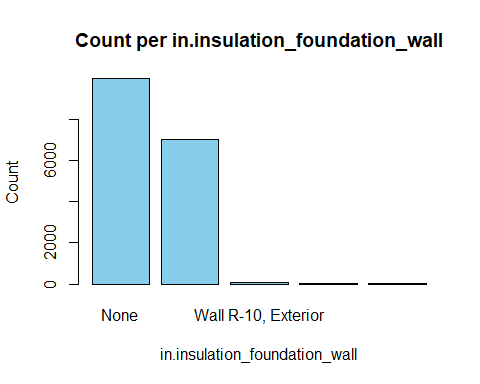
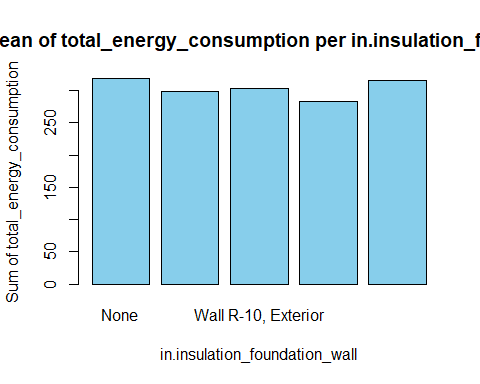
## [1] "in.insulation\_ceiling 46.1538461538462 %"



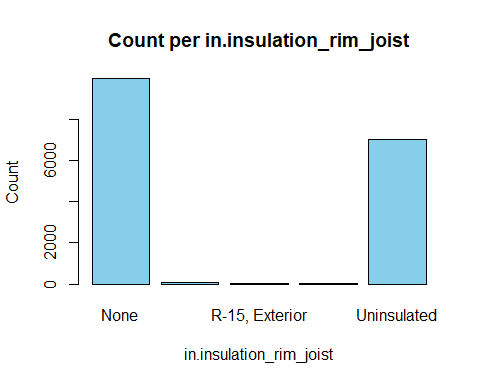
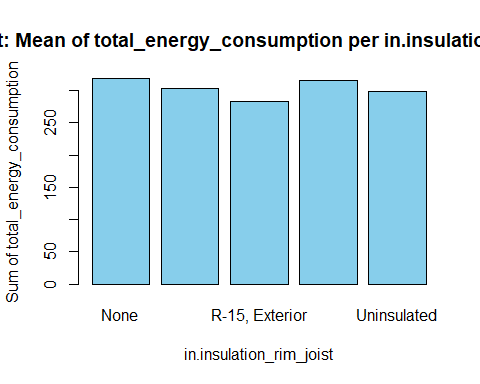
## [1] "in.insulation\_floor 47.2527472527472 %"



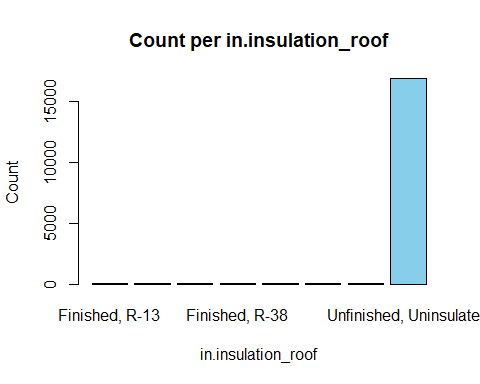
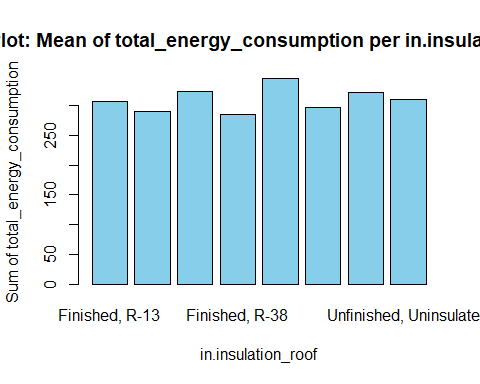
## [1] "in.insulation\_foundation\_wall 48.3516483516484 %"



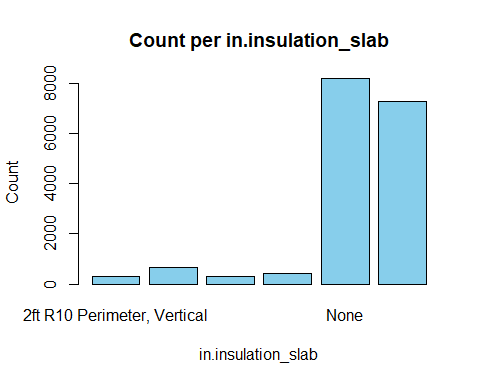
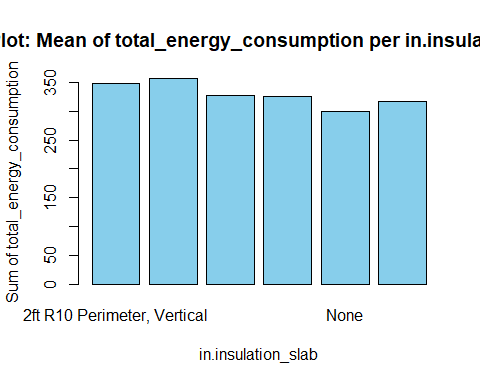
## [1] "in.insulation\_rim\_joist 49.4505494505495 %"



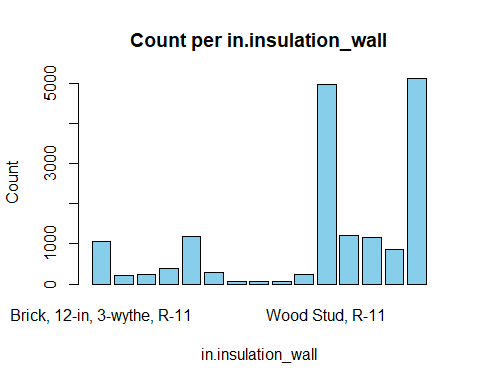
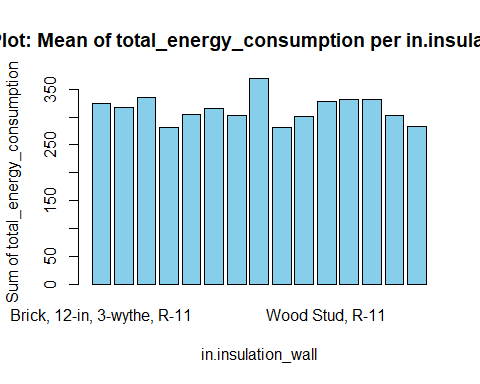
## [1] "in.insulation\_roof 50.5494505494505 %"



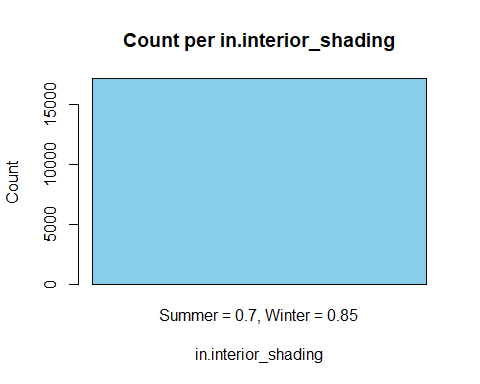
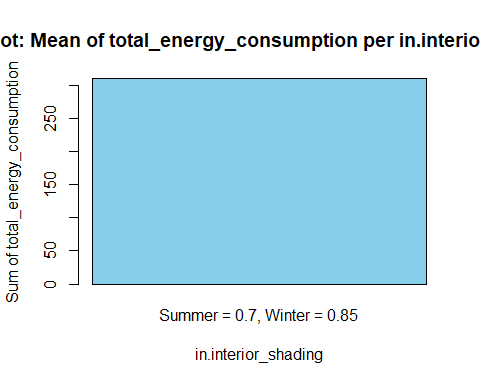
## [1] "in.insulation\_slab 51.6483516483517 %"



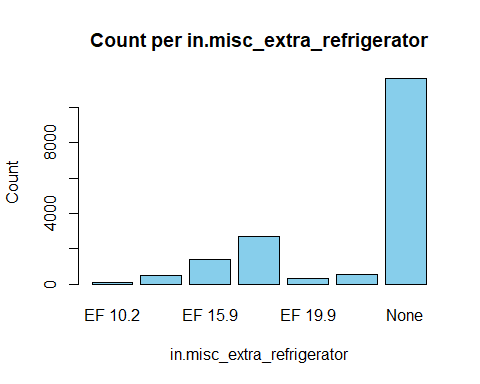
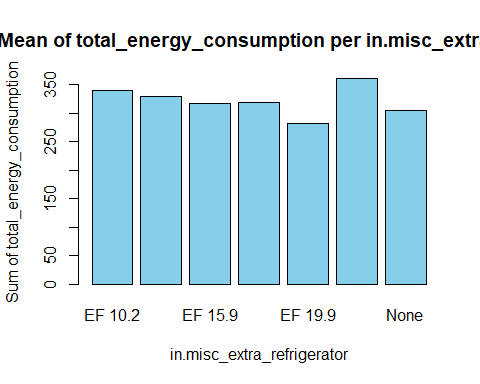
## [1] "in.insulation\_wall 52.7472527472528 %"



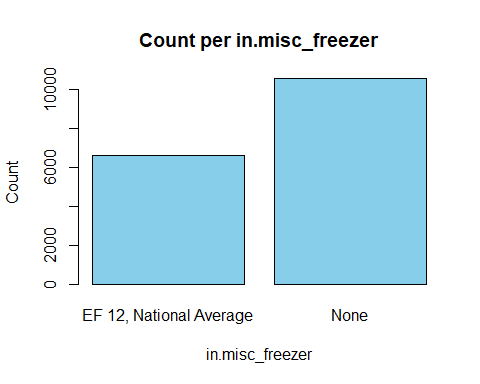
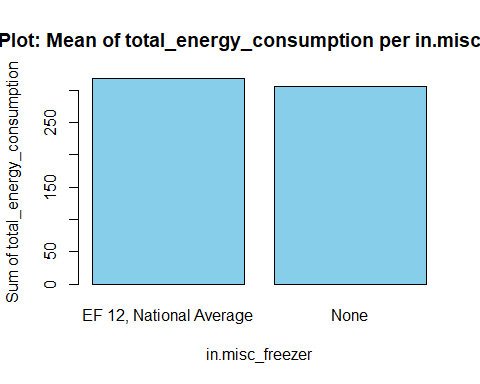
## [1] "in.interior\_shading 53.8461538461538 %"



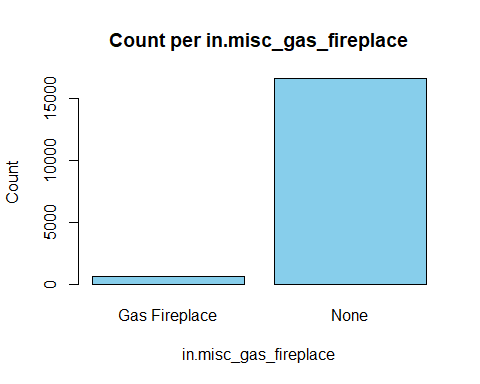
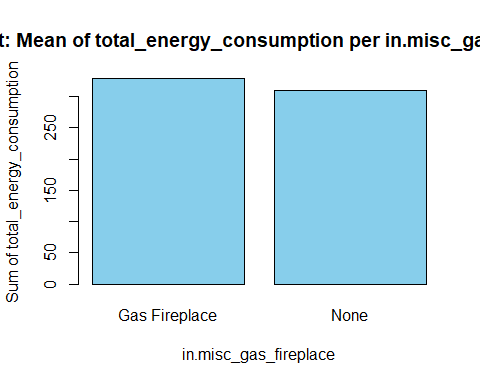
## [1] "in.misc\_extra\_refrigerator 54.9450549450549 %"



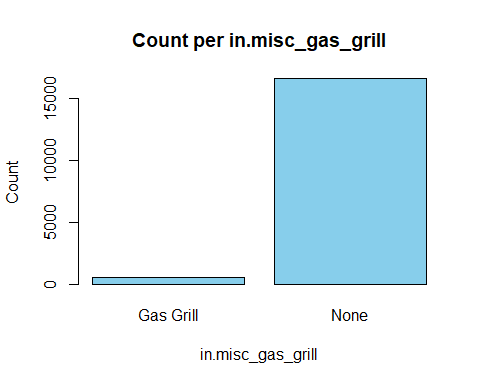
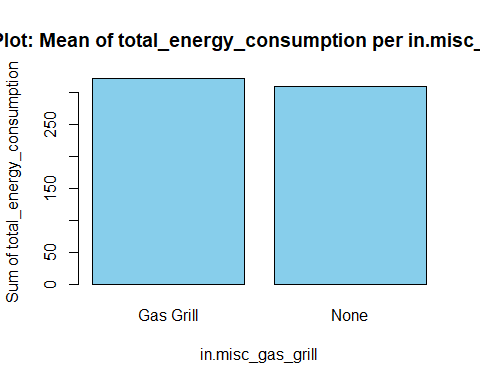
## [1] "in.misc\_freezer 56.043956043956 %"



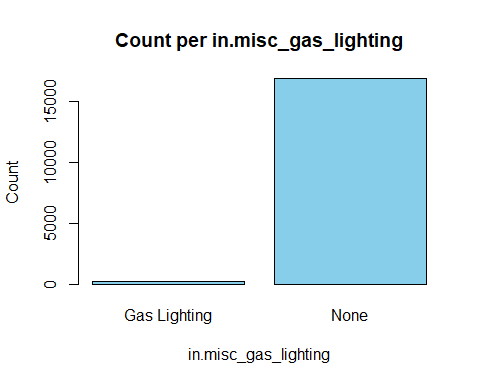
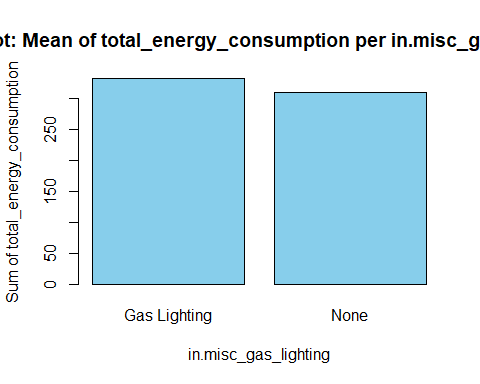
## [1] "in.misc\_gas\_fireplace 57.1428571428571 %"



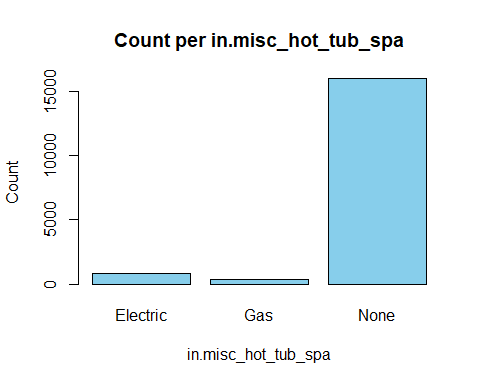
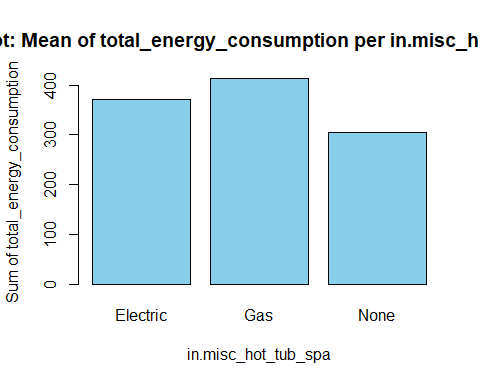
## [1] "in.misc\_gas\_grill 58.2417582417582 %"



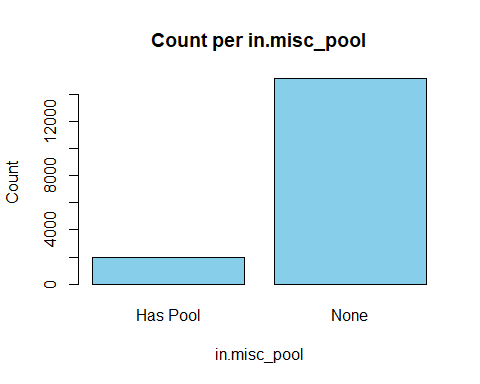
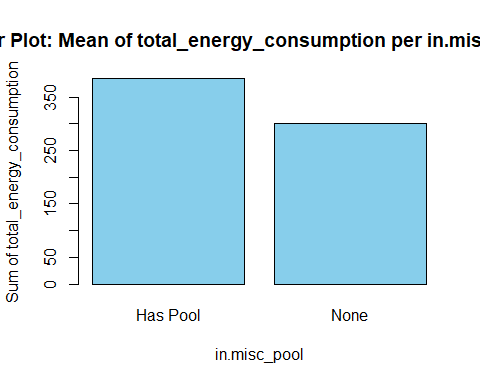
## [1] "in.misc\_gas\_lighting 59.3406593406593 %"



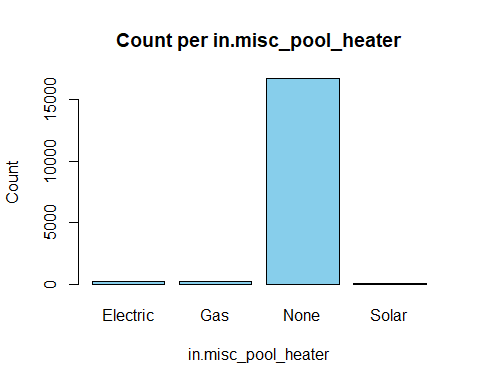
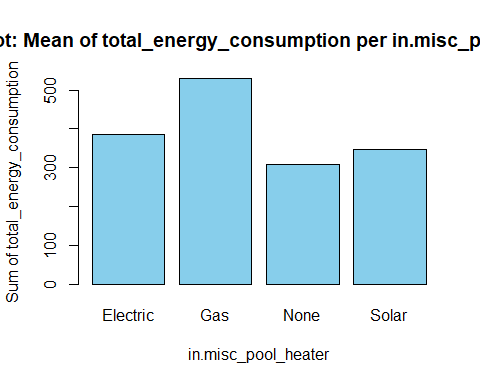
## [1] "in.misc\_hot\_tub\_spa 60.4395604395604 %"



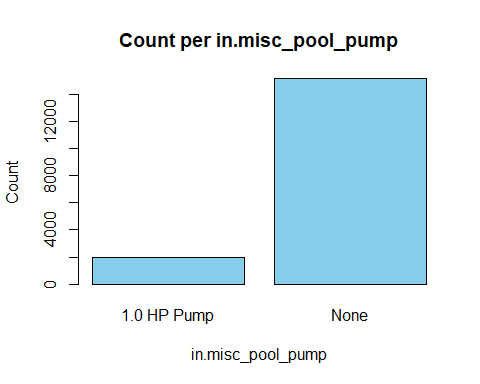
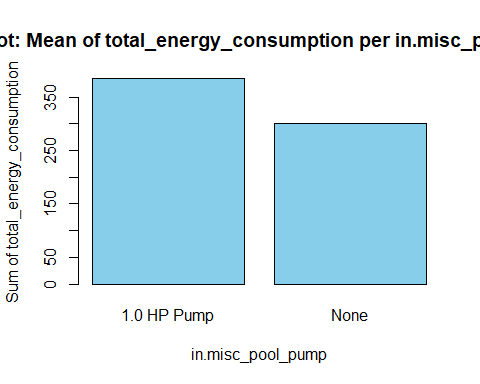
## [1] "in.misc\_pool 61.5384615384615 %"



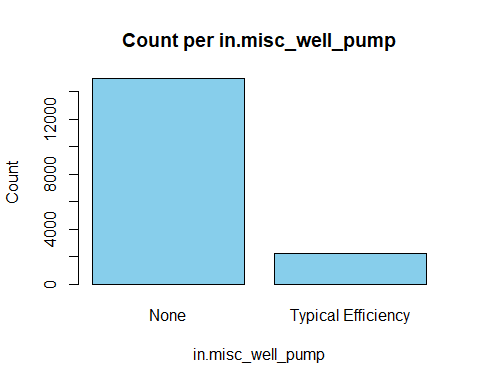
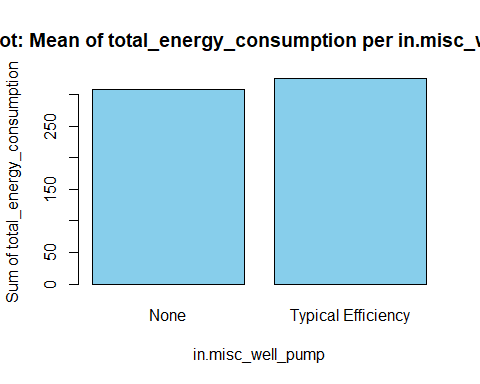
## [1] "in.misc\_pool\_heater 62.6373626373626 %"



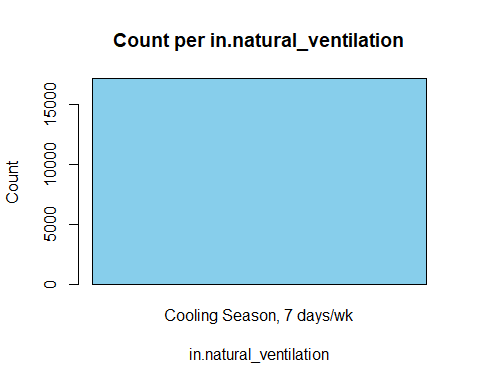
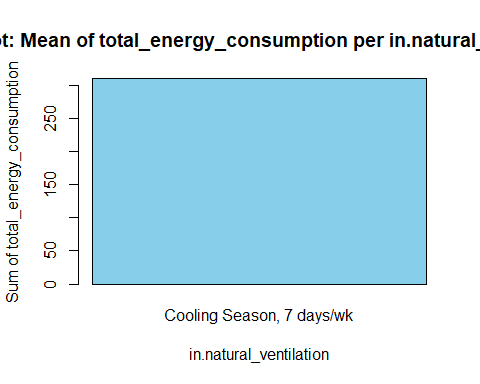
## [1] "in.misc\_pool\_pump 63.7362637362637 %"



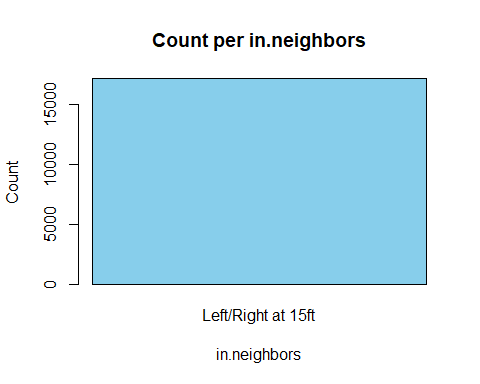
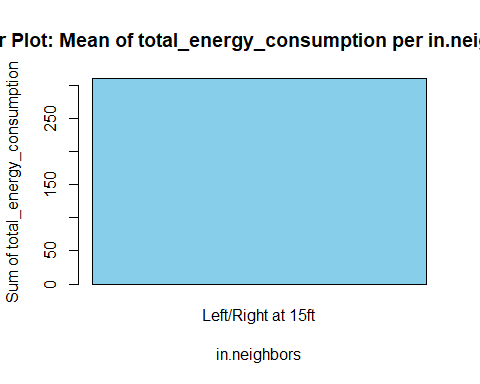
## [1] "in.misc\_well\_pump 64.8351648351648 %"



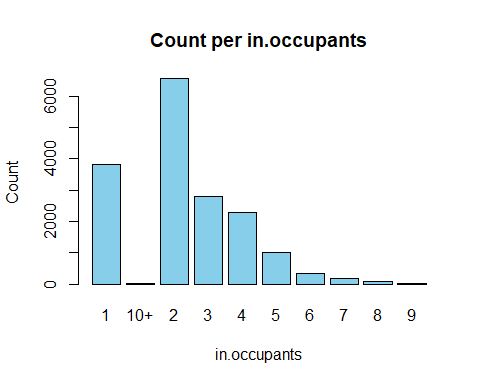
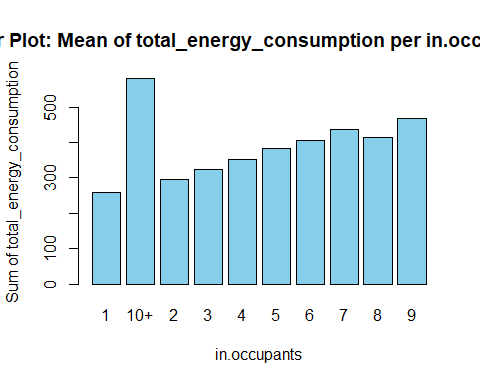
## [1] "in.natural\_ventilation 65.9340659340659 %"



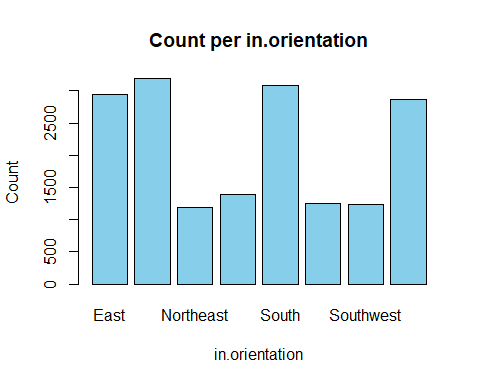
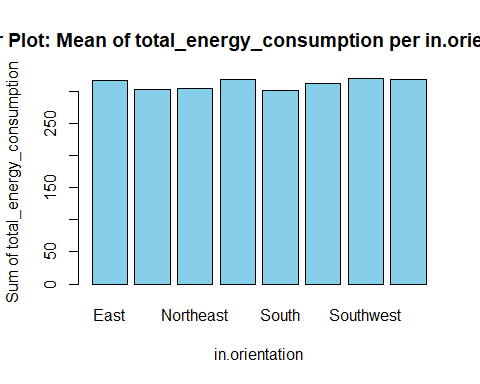
## [1] "in.neighbors 67.032967032967 %"



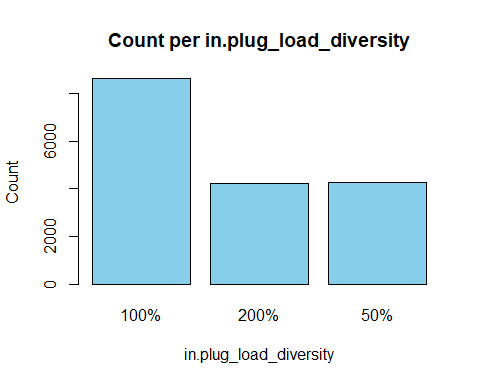
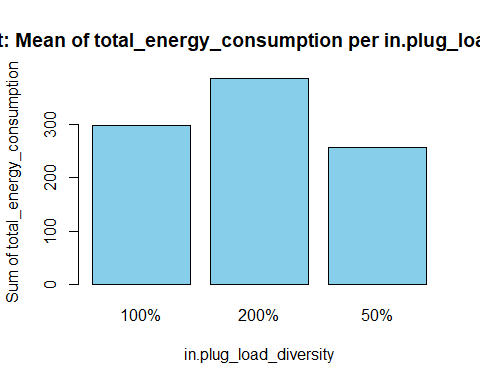
## [1] "in.occupants 68.1318681318681 %"



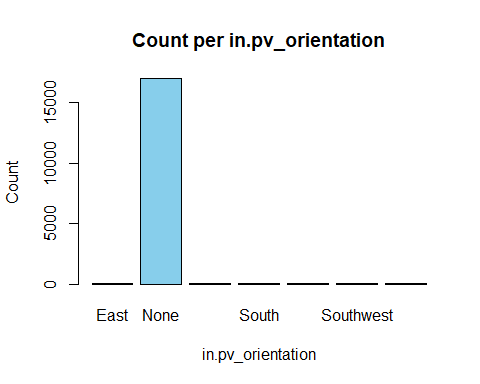
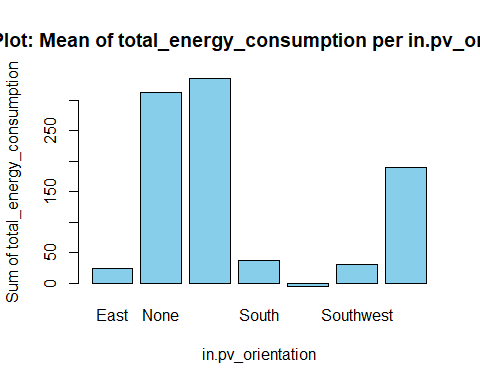
## [1] "in.orientation 69.2307692307692 %"



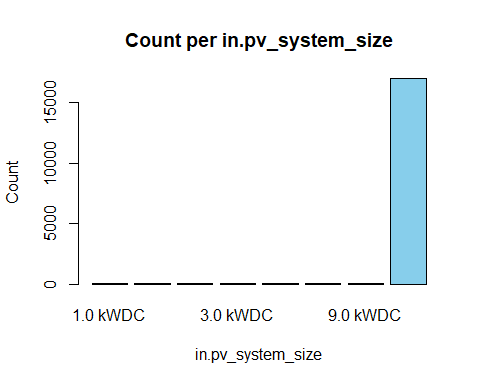
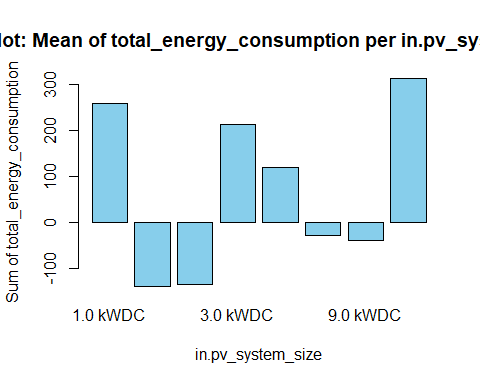
## [1] "in.plug\_load\_diversity 70.3296703296703 %"



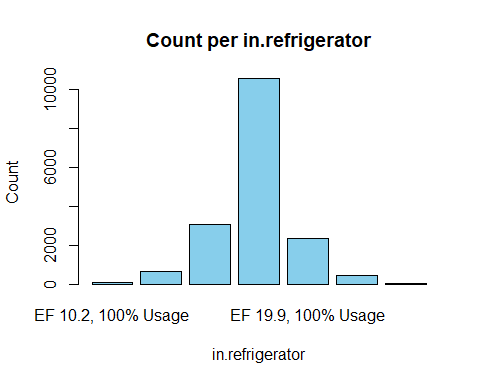
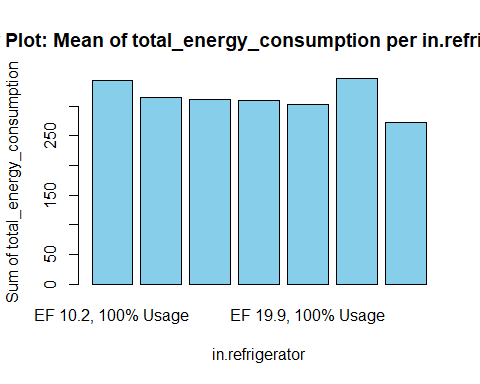
## [1] "in.pv\_orientation 71.4285714285714 %"



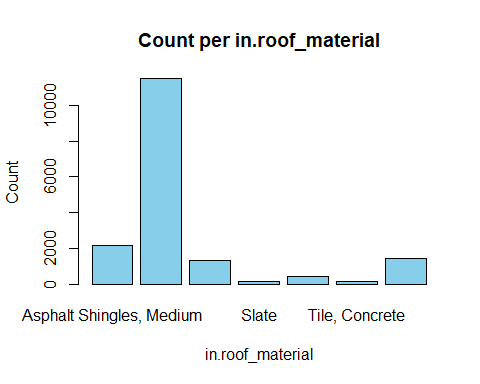
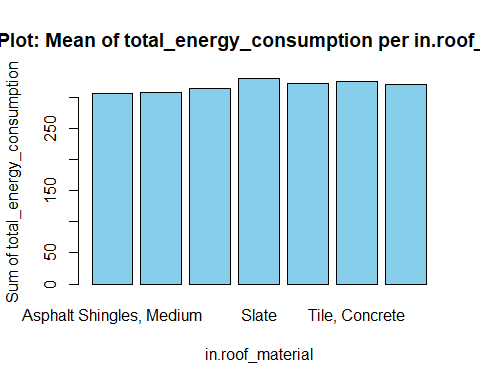
## [1] "in.pv\_system\_size 72.5274725274725 %"



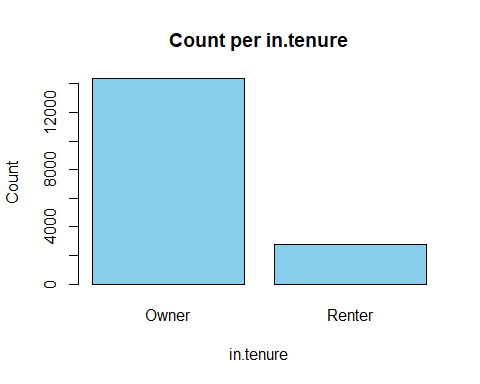
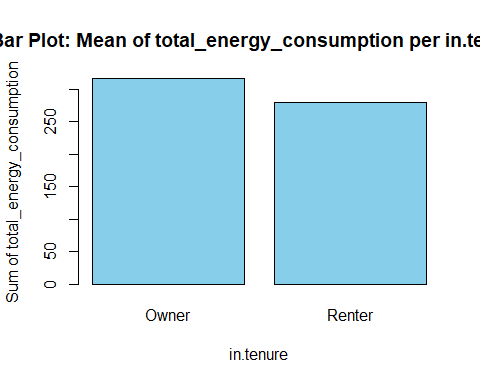
## [1] "in.refrigerator 73.6263736263736 %"



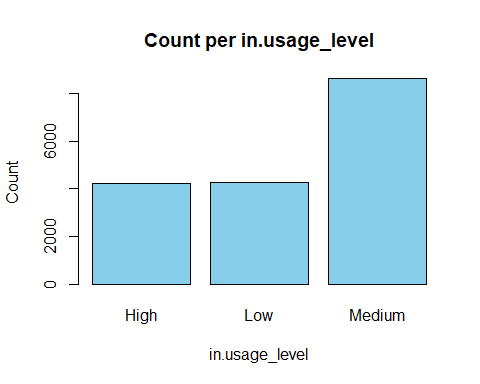
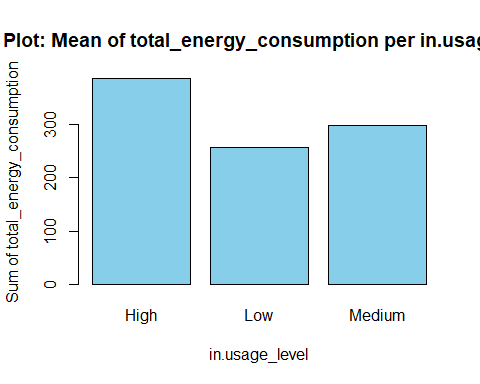
## [1] "in.roof\_material 74.7252747252747 %"



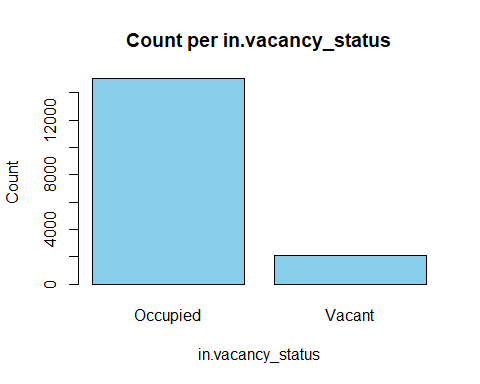
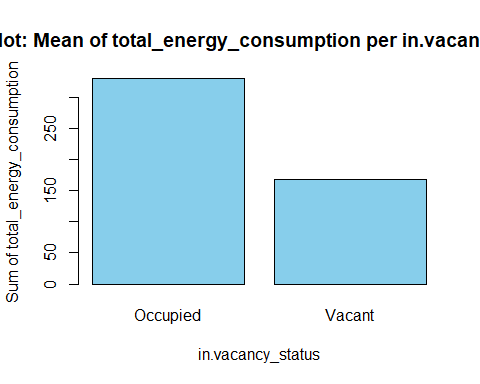
## [1] "in.tenure 75.8241758241758 %"



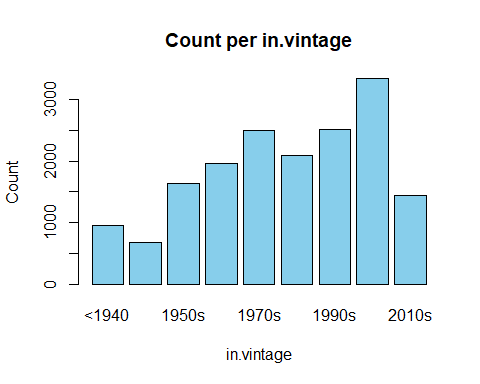
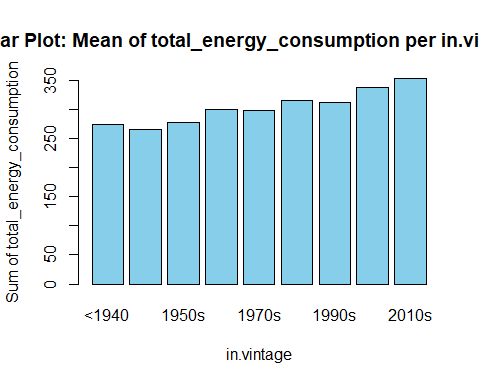
## [1] "in.usage\_level 76.9230769230769 %"



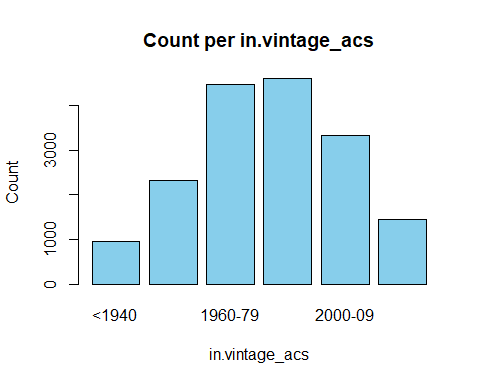
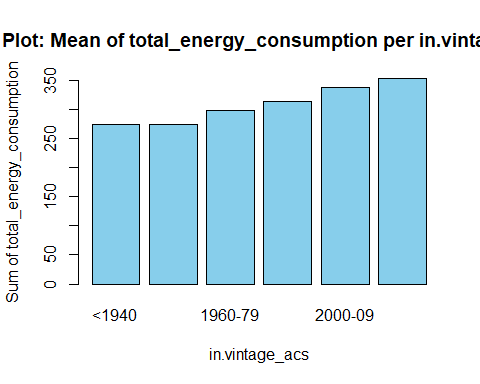
## [1] "in.vacancy\_status 78.021978021978 %"



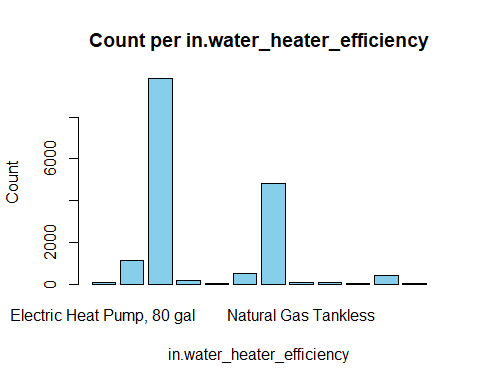
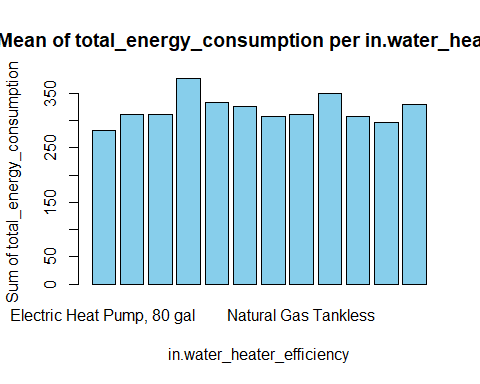
## [1] "in.vintage 79.1208791208791 %"



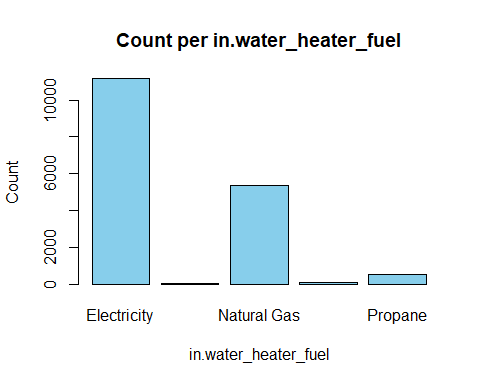
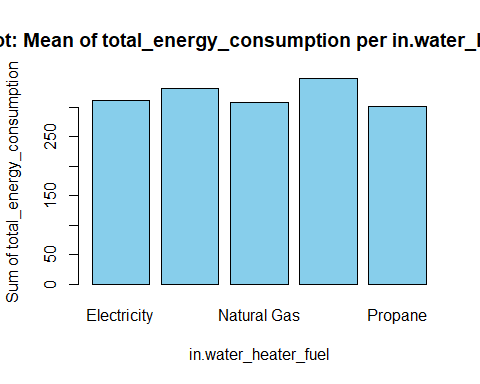
## [1] "in.vintage\_acs 80.2197802197802 %"



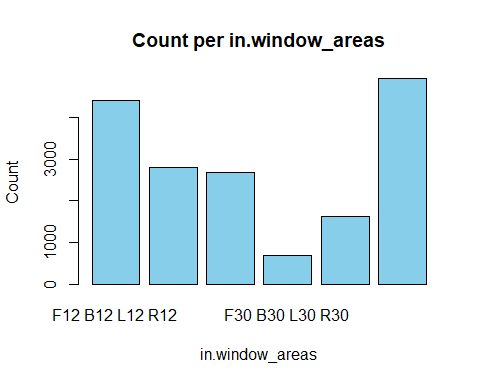
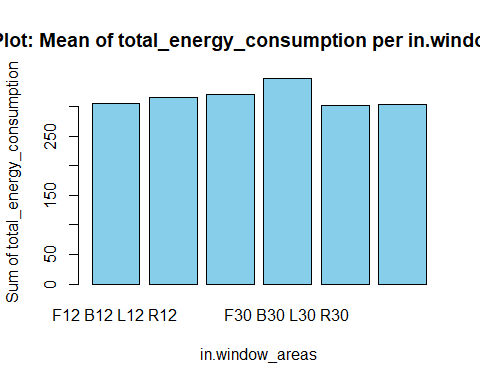
## [1] "in.water\_heater\_efficiency 81.3186813186813 %"



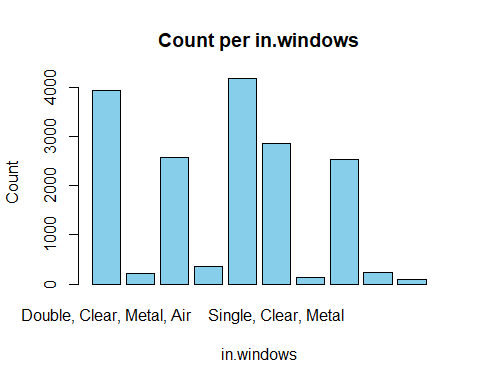
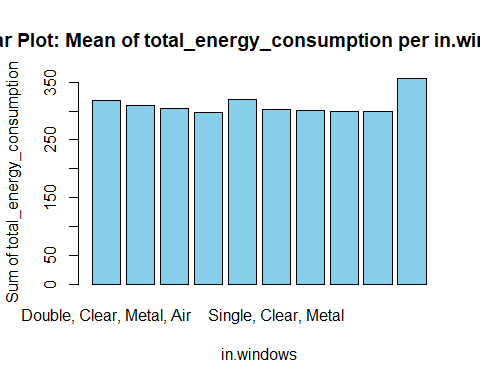
## [1] "in.water\_heater\_fuel 82.4175824175824 %"



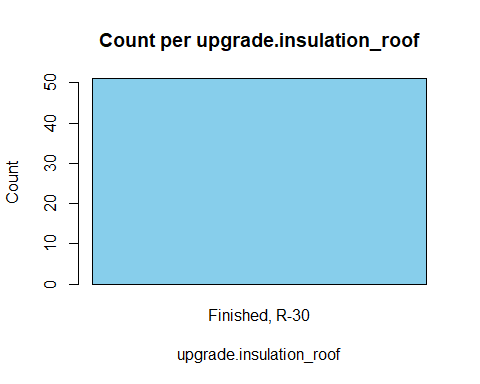
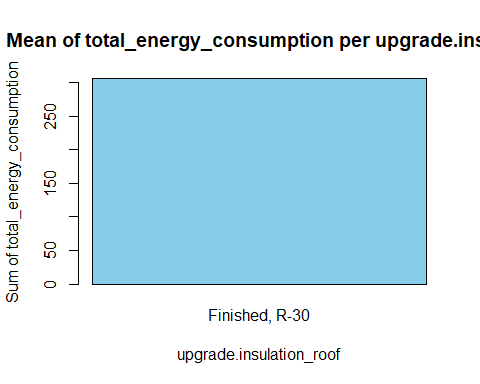
## [1] "in.window\_areas 83.5164835164835 %"



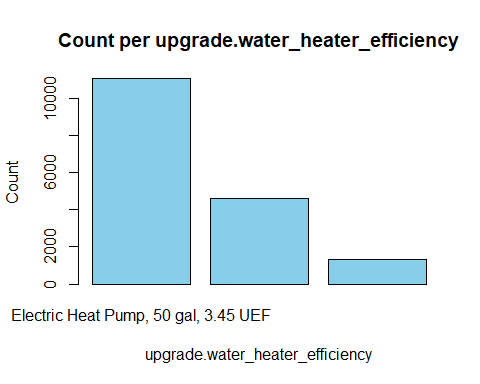
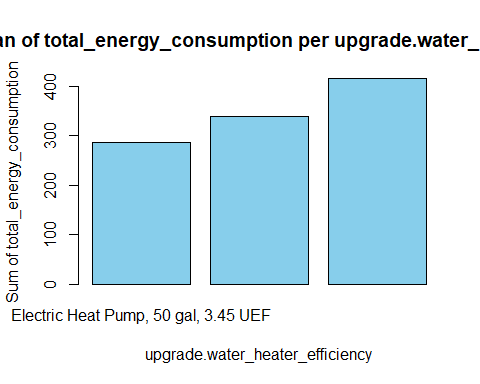
## [1] "in.windows 84.6153846153846 %"



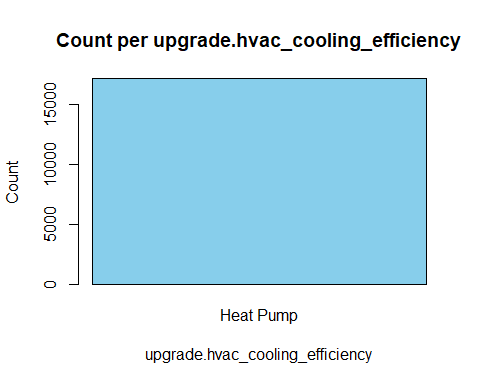
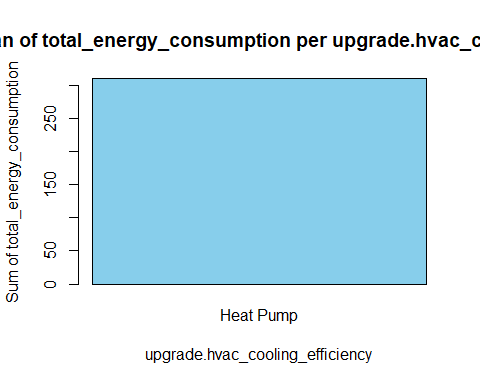
## [1] "upgrade.insulation\_roof 85.7142857142857 %"



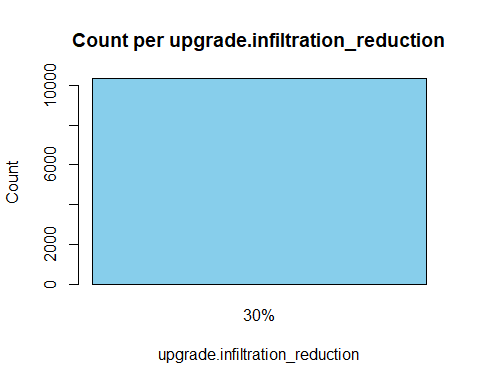
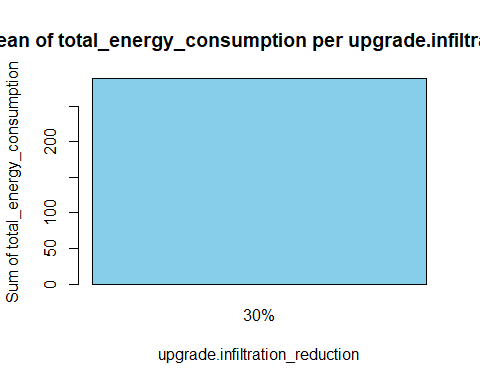
## [1] "upgrade.water\_heater\_efficiency 86.8131868131868 %"



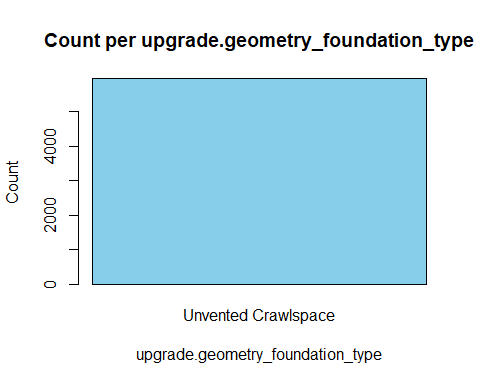
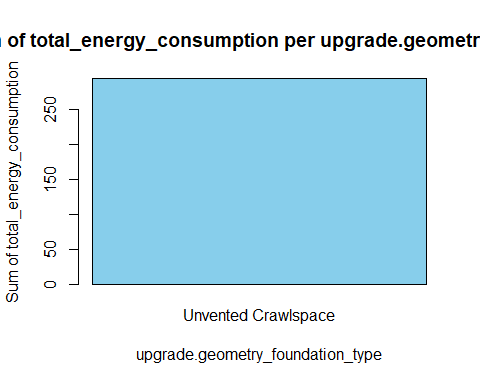
## [1] "upgrade.hvac\_cooling\_efficiency 87.9120879120879 %"



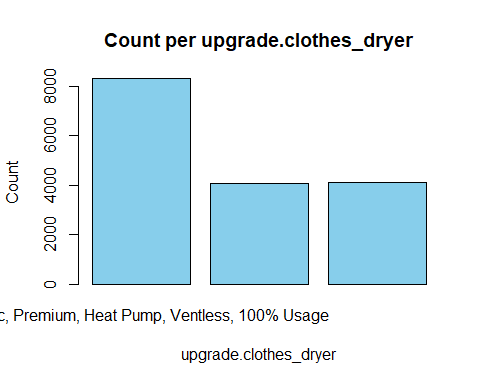
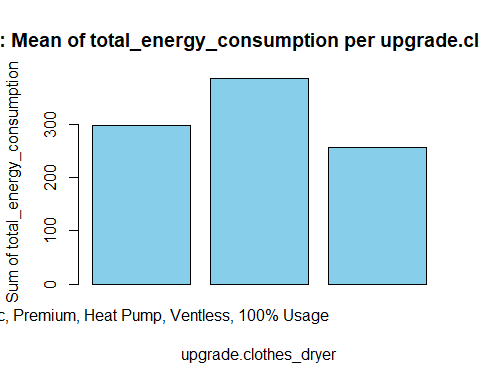
## [1] "upgrade.infiltration\_reduction 89.010989010989 %"



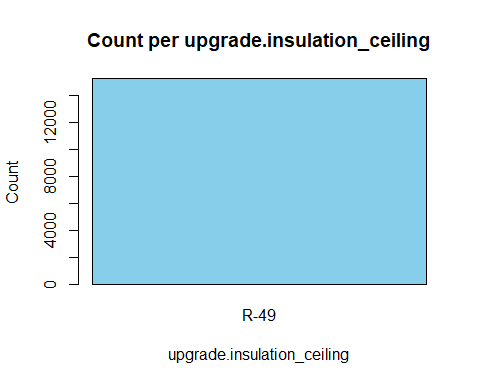
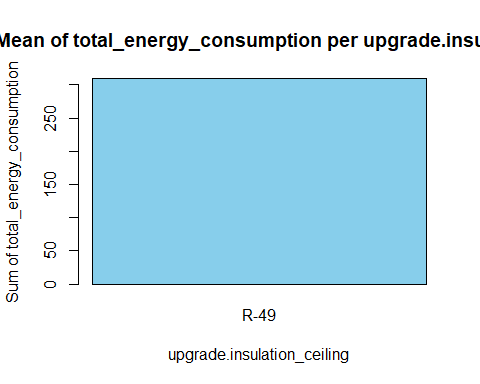
## [1] "upgrade.geometry\_foundation\_type 90.1098901098901 %"



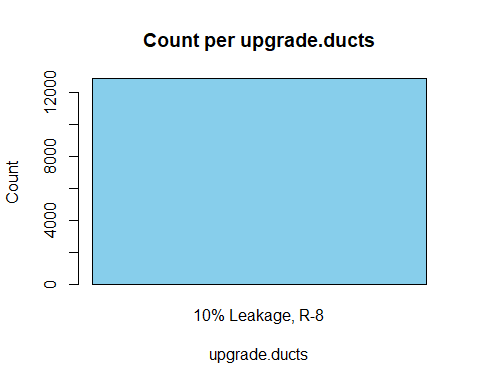
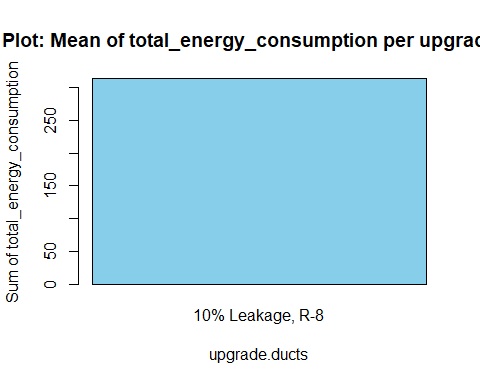
## [1] "upgrade.clothes\_dryer 91.2087912087912 %"



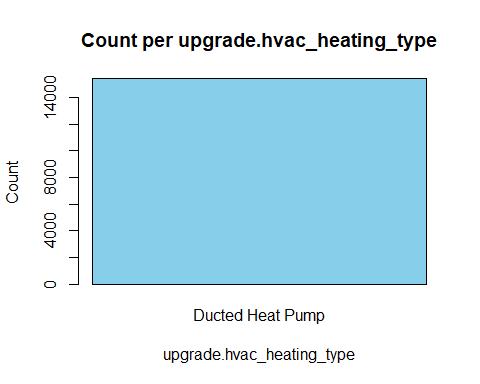
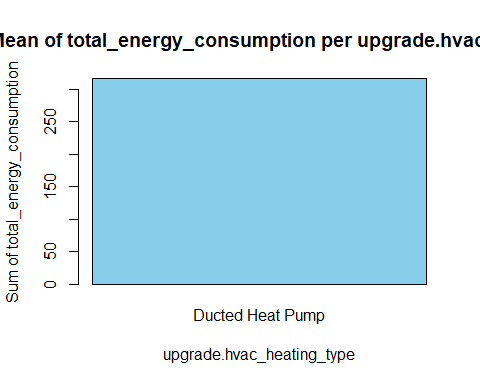
## [1] "upgrade.insulation\_ceiling 92.3076923076923 %"



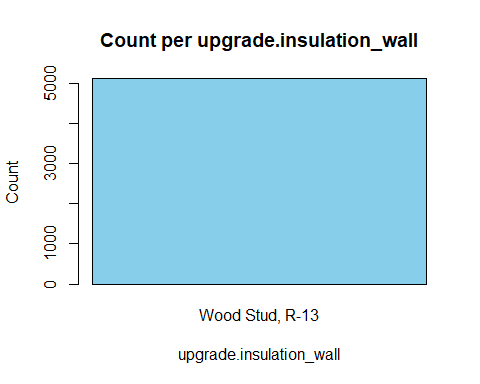
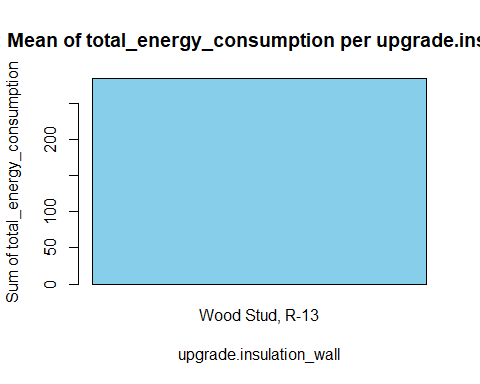
## [1] "upgrade.ducts 93.4065934065934 %"



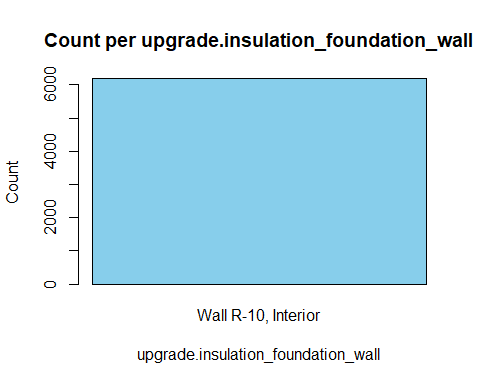
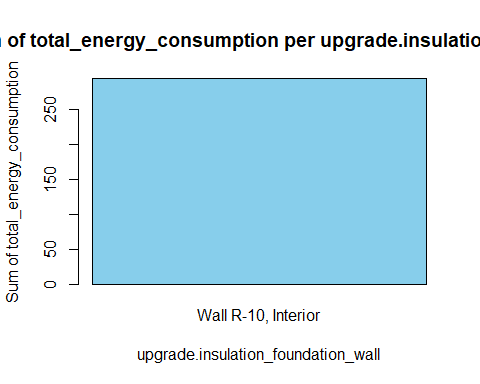
## [1] "upgrade.hvac\_heating\_type 94.5054945054945 %"



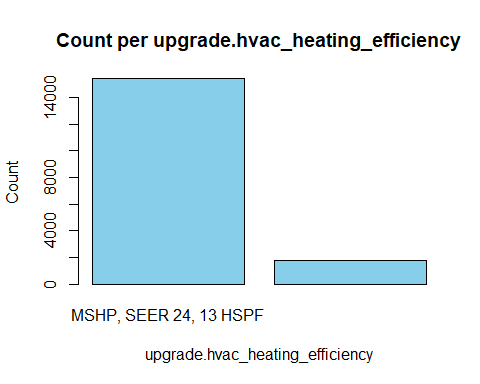
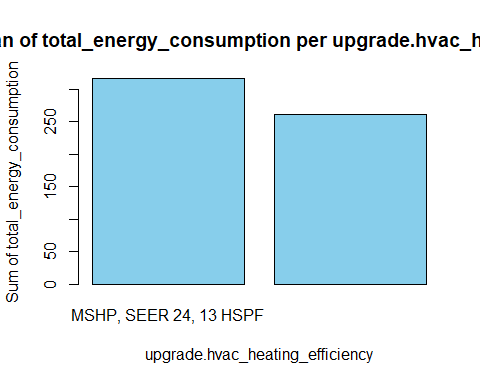
## [1] "upgrade.insulation\_wall 95.6043956043956 %"



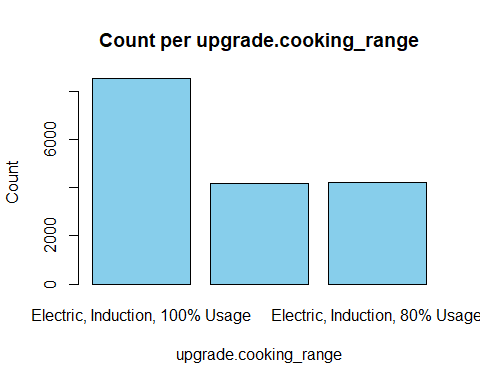
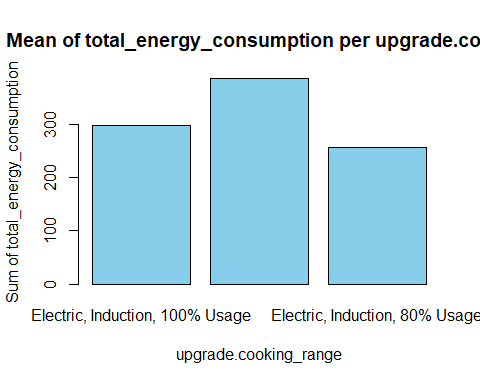
## [1] "upgrade.insulation\_foundation\_wall 96.7032967032967 %"



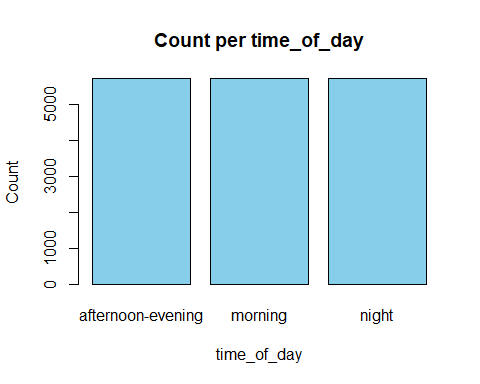
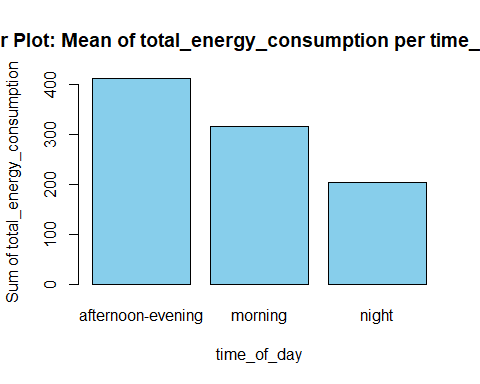
## [1] "upgrade.hvac\_heating\_efficiency 97.8021978021978 %"



## [1] "upgrade.cooking\_range 98.9010989010989 %"



## [1] "time\_of\_day 100 %"



# Handling Null Values  
  
null\_values\_count <- sapply(filtered\_data, function(x) sum(is.na(x)))  
null\_values\_count[null\_values\_count>0]

## upgrade.insulation\_roof upgrade.water\_heater\_efficiency   
## 17079 156   
## upgrade.infiltration\_reduction upgrade.geometry\_foundation\_type   
## 6798 11172   
## upgrade.clothes\_dryer upgrade.insulation\_ceiling   
## 630 1893   
## upgrade.ducts upgrade.hvac\_heating\_type   
## 4296 1731   
## upgrade.insulation\_wall upgrade.insulation\_foundation\_wall   
## 12009 10947   
## upgrade.cooking\_range   
## 201

# frequency\_table <- table(filtered\_data$in.occupants)  
# most\_frequent\_value <- names(frequency\_table[which.max(frequency\_table)])  
  
# replacing null value with the most frequent value  
filtered\_data$upgrade.water\_heater\_efficiency[is.na(filtered\_data$upgrade.water\_heater\_efficiency)] <- 'Electric Heat Pump, 50 gal, 3.45 UEF'  
filtered\_data$upgrade.cooking\_range[is.na(filtered\_data$upgrade.cooking\_range)] <- 'Electric, Induction, 100% Usage'  
  
filtered\_data$in.occupants[is.na(filtered\_data$in.occupants)] <- 1  
  
# removing columns with high null values(> 500)  
columns\_to\_remove <- c("upgrade.insulation\_roof", "upgrade.geometry\_foundation\_type", "upgrade.insulation\_wall", "upgrade.insulation\_foundation\_wall","upgrade.infiltration\_reduction", "upgrade.clothes\_dryer", "upgrade.insulation\_ceiling", "upgrade.ducts", "upgrade.hvac\_heating\_type")  
  
filtered\_data <- filtered\_data[, !names(filtered\_data) %in% columns\_to\_remove]

# find number of character variables  
non\_numeric\_cols <- sapply(filtered\_data, function(x) !is.numeric(x) && !is.integer(x))  
non\_num\_col\_names <- names(filtered\_data)[non\_numeric\_cols] # 86 cols  
char\_cols <- filtered\_data[, non\_num\_col\_names]  
  
# converting in.occupants to numeric  
filtered\_data$in.occupants <- as.numeric(filtered\_data$in.occupants)

## Warning: NAs introduced by coercion

# find variables with just 1 unique value  
unique\_counts <- list()  
for(column\_name in names(char\_cols)) {  
 if(is.factor(char\_cols[[column\_name]]) || is.character(char\_cols[[column\_name]])) {  
 unique\_counts[[column\_name]] <- length(unique(char\_cols[[column\_name]]))  
 }  
}  
  
# removing variables having only 1 unique value  
columns\_to\_remove <- names(unique\_counts[unique\_counts == 1])  
filtered\_data <- filtered\_data[, !names(filtered\_data) %in% columns\_to\_remove]  
  
for(column\_name in names(df)) {  
 df[[column\_name]] <- as.factor(df[[column\_name]])  
}

# One-hot-encoding  
  
library(caret)

## Loading required package: lattice

##   
## Attaching package: 'caret'

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

dummy\_model <- dummyVars("~ .", data = filtered\_data)  
transformed\_data <- predict(dummy\_model, newdata = filtered\_data)  
transformed\_data <- as.data.frame(transformed\_data)

# Correlation Analysis  
#install.packages("ltm")  
#library(ltm)  
library(dplyr)  
#install.packages("polycor")  
library(polycor)

## Warning: package 'polycor' was built under R version 4.3.2

target <- transformed\_data$total\_energy\_consumption  
  
# Initialize an empty vector to store correlations  
correlations <- numeric()  
  
# Initialize an empty vector to store variable names  
variable\_names <- character()  
  
# Loop through columns and calculate correlations  
for (col\_name in names(transformed\_data)) {  
 col <- transformed\_data[[col\_name]]  
   
 if (is.numeric(col)) {  
 # Calculate numeric correlations (Pearson)  
 correlation <- cor(col, target, method = "pearson")  
 } else {  
 # Handle categorical variables (e.g., point-biserial correlation)  
 correlation <- point.biserial(target, col)  
 }  
   
 # Store the correlation value and variable name  
 correlations <- c(correlations, correlation)  
 variable\_names <- c(variable\_names, col\_name)  
}  
  
# Combine correlations with variable names into a data frame  
correlations\_df <- data.frame(Variable = variable\_names, Correlation = correlations)  
  
# Sort correlations by absolute value in descending order  
correlations\_df <- correlations\_df[order(-abs(correlations\_df$Correlation)),]  
  
# View(correlations\_df)  
# Print the top correlated variables  
print(head(correlations\_df, n = 30))

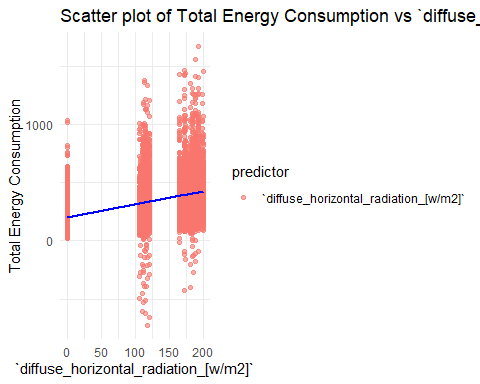
## Variable  
## 517 total\_energy\_consumption  
## 524 `diffuse\_horizontal\_radiation\_[w/m2]`  
## 522 `global\_horizontal\_radiation\_[w/m2]`  
## 518 `dry\_bulb\_temperature\_[°c]`  
## 523 `direct\_normal\_radiation\_[w/m2]`  
## 519 `relative\_humidity\_[%]`  
## 516 time\_of\_daynight  
## 2 in.sqft  
## 514 time\_of\_dayafternoon-evening  
## 520 `wind\_speed\_[m/s]`  
## 152 in.geometry\_floor\_area4000+  
## 158 in.geometry\_floor\_area\_bin4000+  
## 456 in.vacancy\_statusOccupied  
## 457 in.vacancy\_statusVacant  
## 8 in.ceiling\_fanStandard Efficiency, No usage  
## 234 in.hot\_water\_fixtures200% Usage  
## 420 in.plug\_load\_diversity200%  
## 453 in.usage\_levelHigh  
## 512 upgrade.cooking\_rangeElectric, Induction, 120% Usage  
## 26 in.clothes\_dryerElectric, 120% Usage  
## 155 in.geometry\_floor\_area\_bin0-1499  
## 3 in.bedrooms  
## 192 in.heating\_setpoint  
## 521 `wind\_direction\_[deg]`  
## 45 in.cooking\_rangeElectric, 120% Usage  
## 506 upgrade.water\_heater\_efficiencyElectric Heat Pump, 50 gal, 3.45 UEF  
## 54 in.cooling\_setpoint  
## 36 in.clothes\_washerEnergyStar, 120% Usage  
## 147 in.geometry\_floor\_area1000-1499  
## 508 upgrade.water\_heater\_efficiencyElectric Heat Pump, 80 gal, 3.45 UEF  
## Correlation  
## 517 1.0000000  
## 524 0.5060606  
## 522 0.4942252  
## 518 0.4820150  
## 523 0.4700935  
## 519 -0.4539951  
## 516 -0.4498410  
## 2 0.4438015  
## 514 0.4242852  
## 520 0.3769385  
## 152 0.3573836  
## 158 0.3573836  
## 456 0.3182012  
## 457 -0.3182012  
## 8 -0.2658037  
## 234 0.2615688  
## 420 0.2615688  
## 453 0.2615688  
## 512 0.2580684  
## 26 0.2509297  
## 155 -0.2461148  
## 3 0.2415052  
## 192 0.2310107  
## 521 0.2298198  
## 45 0.2065848  
## 506 -0.1972546  
## 54 -0.1916000  
## 36 0.1896092  
## 147 -0.1823668  
## 508 0.1785182

#write.csv(filtered\_data, "C:/Users/Deep/Desktop/Sem-1 SYR/Intro to DS Lab/Project/clean\_data.csv")

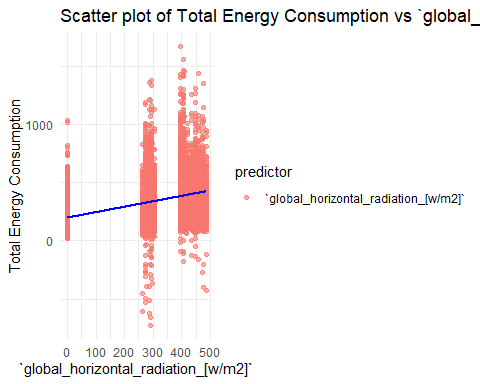
# List of predictor columns  
predictors <- c("`diffuse\_horizontal\_radiation\_[w/m2]`",   
 "`global\_horizontal\_radiation\_[w/m2]`",   
 "`dry\_bulb\_temperature\_[°c]`",   
 "`direct\_normal\_radiation\_[w/m2]`",   
 "`relative\_humidity\_[%]`",  
 "`wind\_speed\_[m/s]`",  
 "in.sqft",  
 "in.heating\_setpoint",  
 "in.cooling\_setpoint",  
 "in.bedrooms",  
 "in.heating\_setpoint",  
 "in.cooling\_setpoint")  
  
# Loop through each predictor and create a scatter plot  
for (predictor in predictors) {  
 plot <- ggplot(filtered\_data, aes\_string(x = predictor, y = "total\_energy\_consumption")) +  
 geom\_point(aes(color = predictor), alpha = 0.6) + # Add color and set transparency  
 geom\_smooth(method = "lm", se = FALSE, color = "blue") + # Add linear regression line  
 labs(title = paste("Scatter plot of Total Energy Consumption vs", predictor),  
 x = predictor,  
 y = "Total Energy Consumption") +  
 theme\_minimal()  
   
 print(plot)  
}

## Warning: `aes\_string()` was deprecated in ggplot2 3.0.0.  
## ℹ Please use tidy evaluation idioms with `aes()`.  
## ℹ See also `vignette("ggplot2-in-packages")` for more information.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

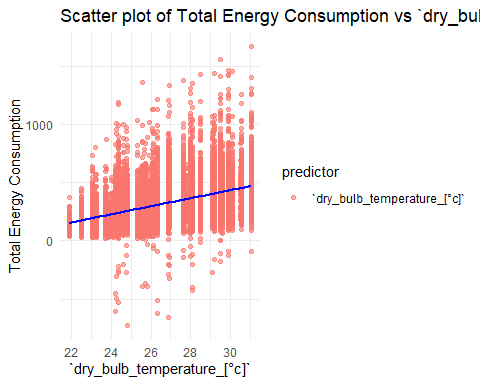
## `geom\_smooth()` using formula = 'y ~ x'



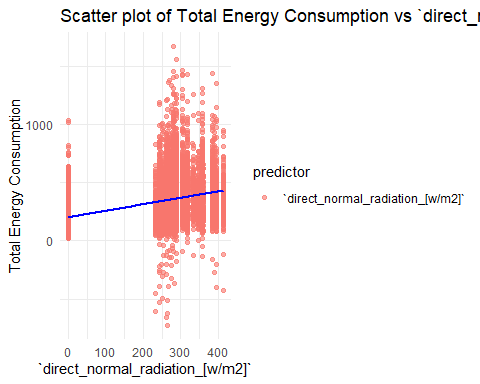
## `geom\_smooth()` using formula = 'y ~ x'



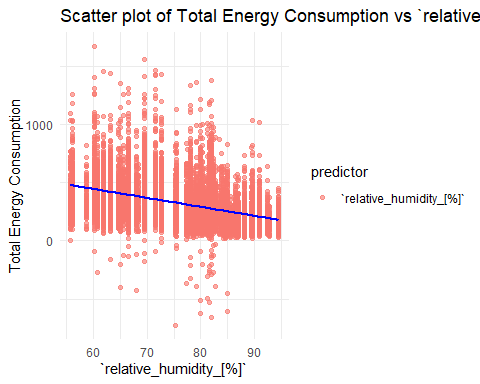
## `geom\_smooth()` using formula = 'y ~ x'



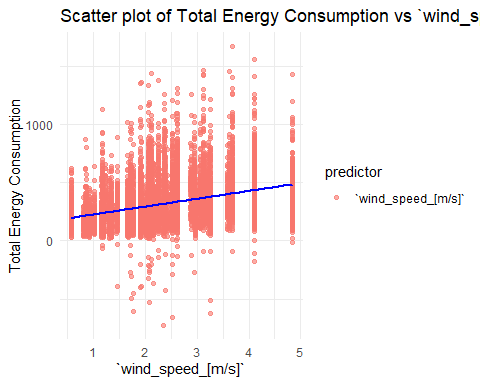
## `geom\_smooth()` using formula = 'y ~ x'



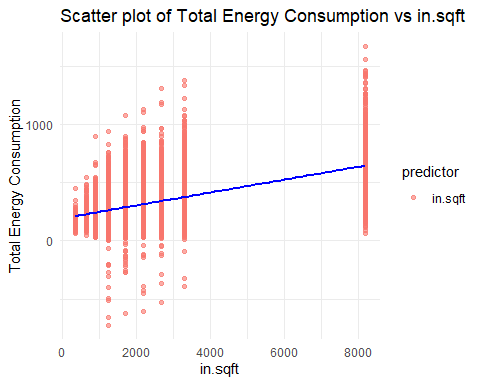
## `geom\_smooth()` using formula = 'y ~ x'



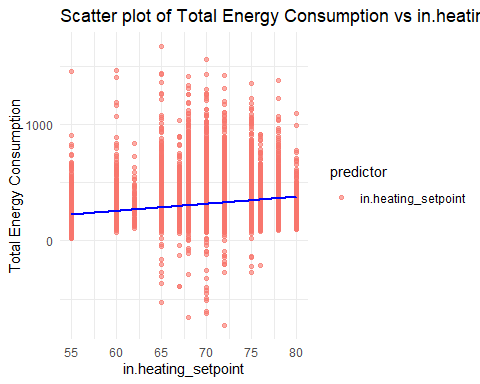
## `geom\_smooth()` using formula = 'y ~ x'



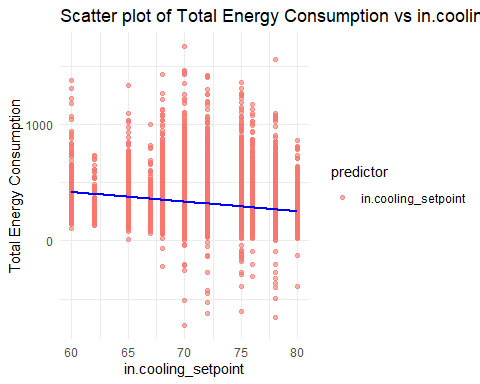
## `geom\_smooth()` using formula = 'y ~ x'



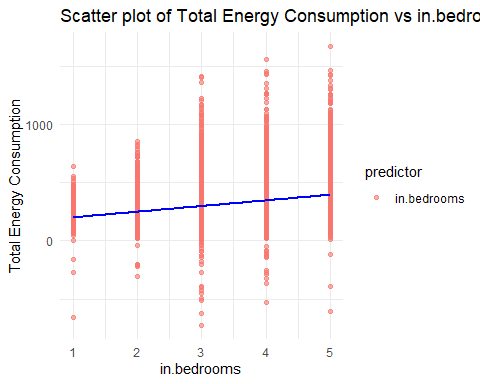
## `geom\_smooth()` using formula = 'y ~ x'



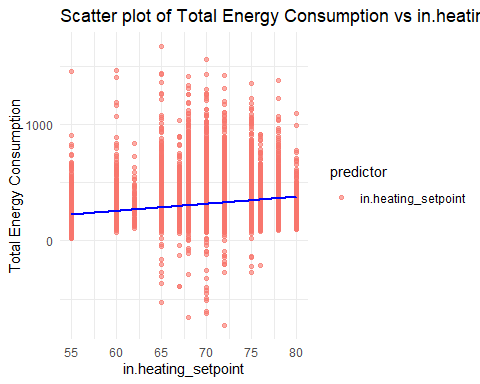
## `geom\_smooth()` using formula = 'y ~ x'



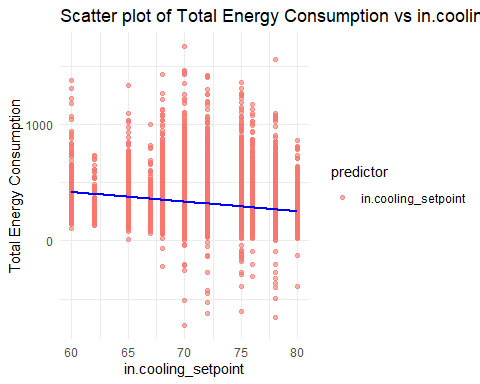
## `geom\_smooth()` using formula = 'y ~ x'



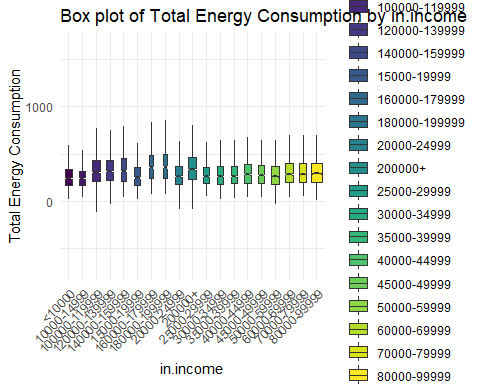
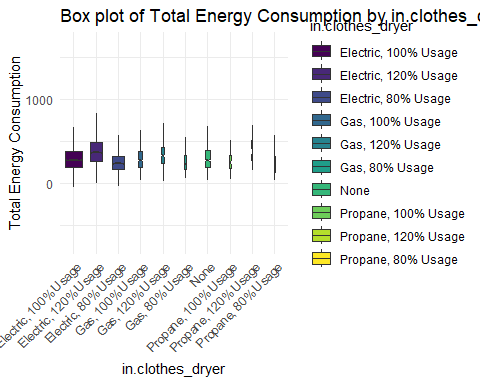
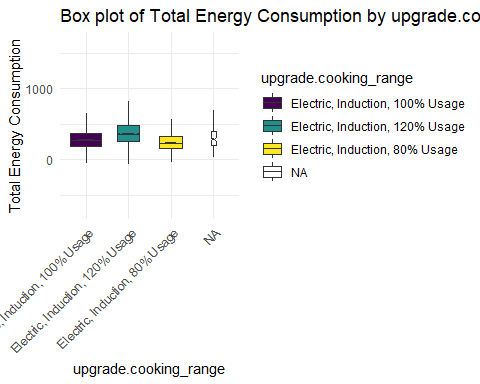
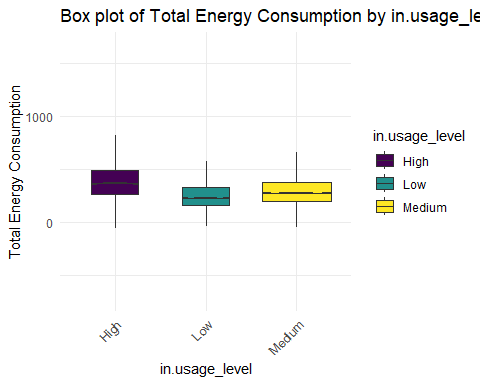
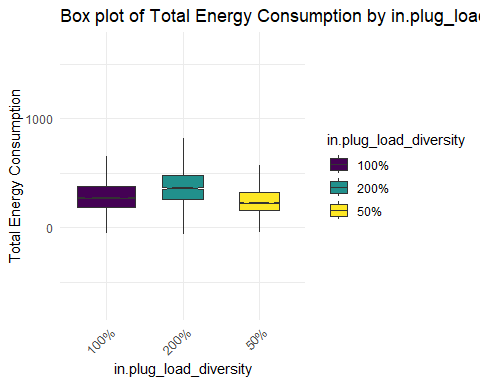
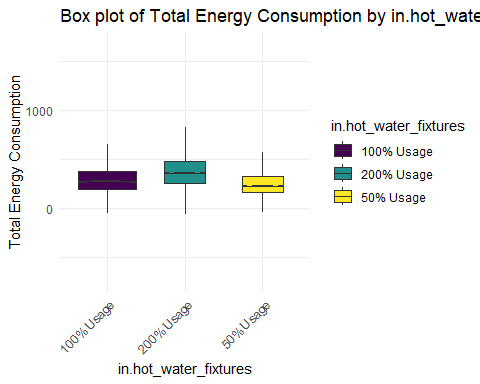
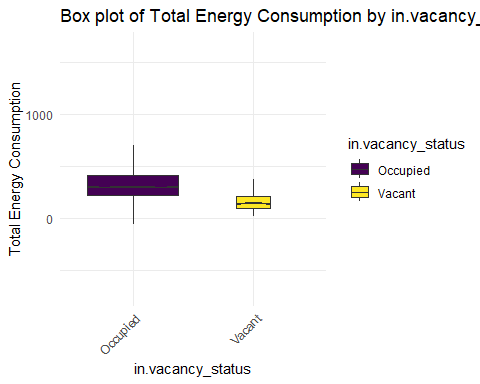
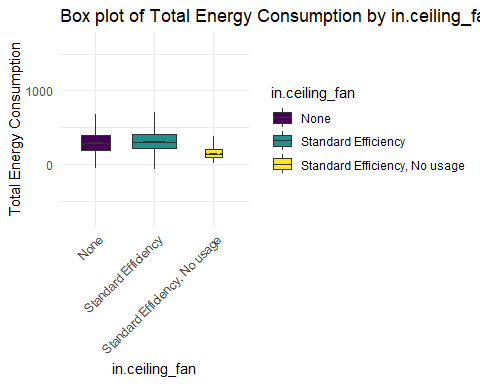
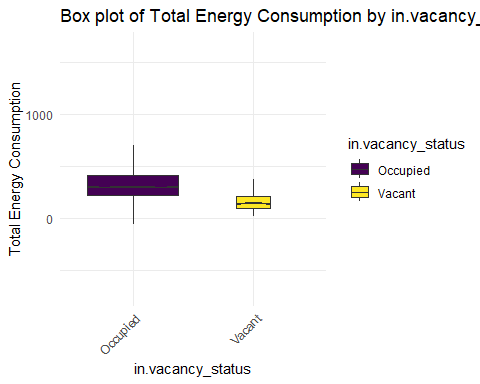
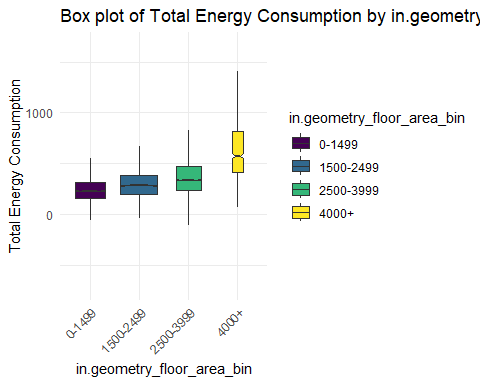
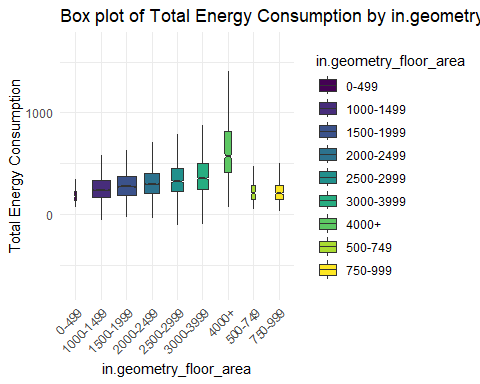
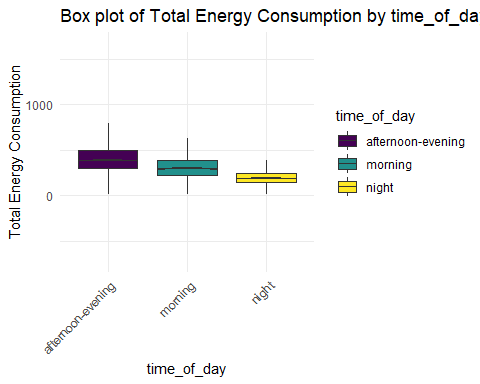
## `geom\_smooth()` using formula = 'y ~ x'



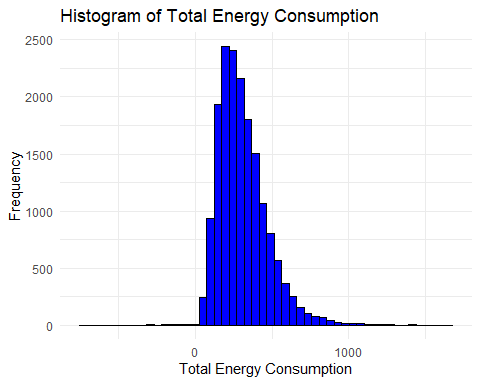
## `geom\_smooth()` using formula = 'y ~ x'



predictors <- c("time\_of\_day",  
 "in.geometry\_floor\_area",  
 "in.geometry\_floor\_area\_bin",  
 "in.vacancy\_status",  
 "in.ceiling\_fan",  
 "in.vacancy\_status",  
 "in.hot\_water\_fixtures",  
 "in.plug\_load\_diversity",  
 "in.usage\_level",  
 "upgrade.cooking\_range",  
 "in.clothes\_dryer",  
 "in.income")  
  
for (predictor in predictors) {  
 plot <- ggplot(data, aes\_string(x = predictor, y = "total\_energy\_consumption", fill = predictor)) +  
 geom\_boxplot(outlier.shape = NA, varwidth = TRUE, notch = TRUE) + # Varying width and notch  
 scale\_fill\_viridis\_d() + # Use Viridis color palette  
 labs(title = paste("Box plot of Total Energy Consumption by", predictor),  
 x = predictor,  
 y = "Total Energy Consumption") +  
 theme\_minimal() +   
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))  
   
 print(plot)  
}



ggplot(data, aes(x = total\_energy\_consumption)) +  
 geom\_histogram(bins = 50, fill = "blue", color = "black") +  
 labs(title = "Histogram of Total Energy Consumption",  
 x = "Total Energy Consumption",  
 y = "Frequency") +  
 theme\_minimal()



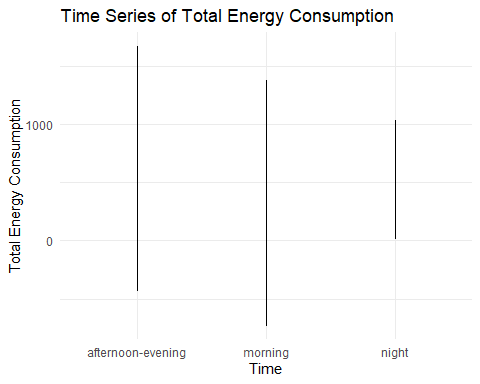
library(corrplot)

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

## corrplot 0.92 loaded

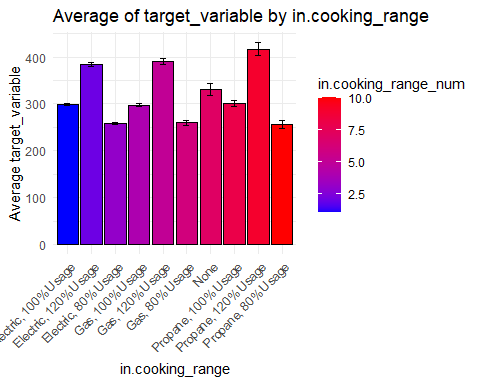
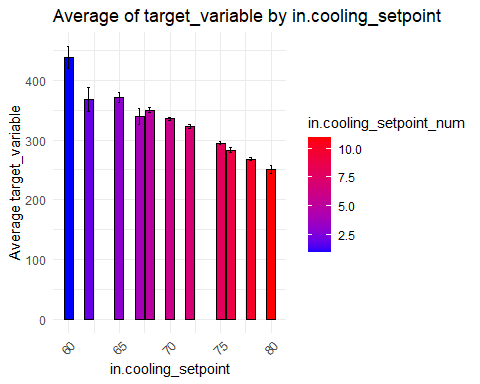
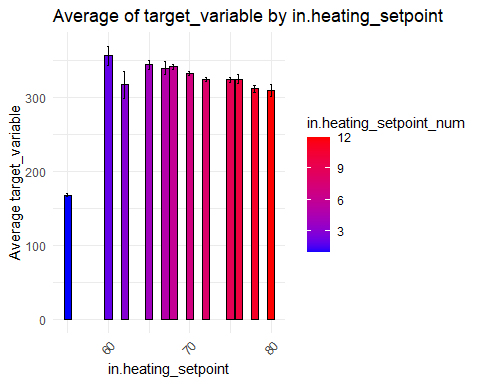
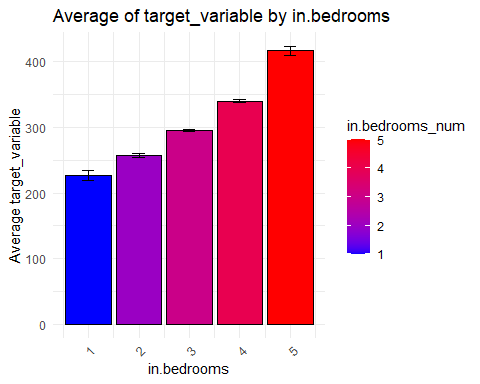
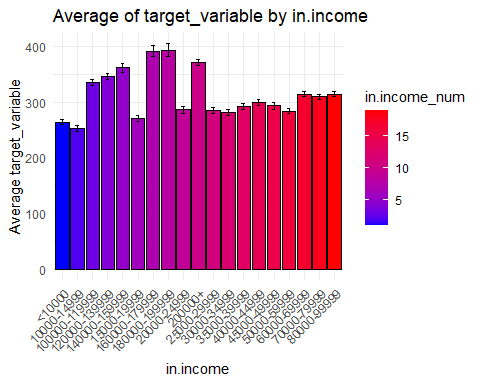
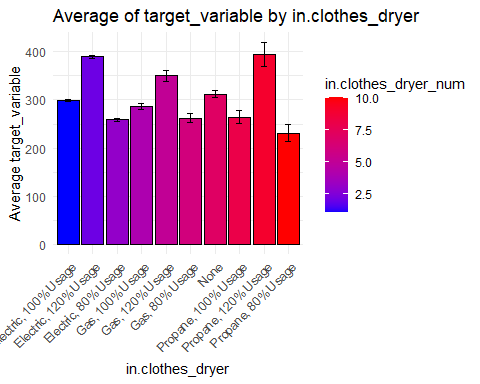
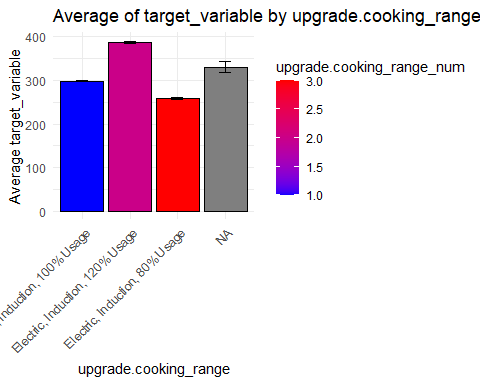
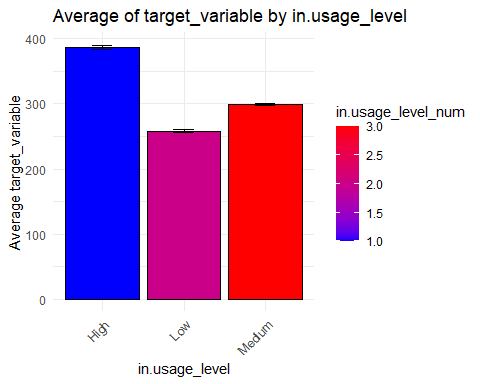
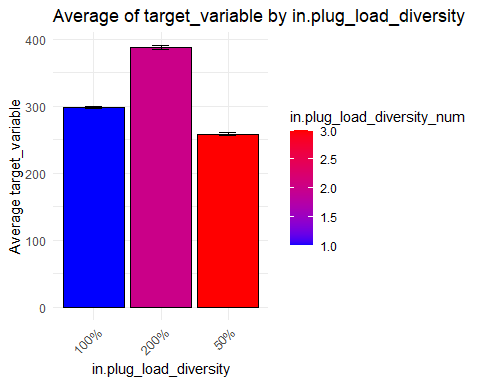
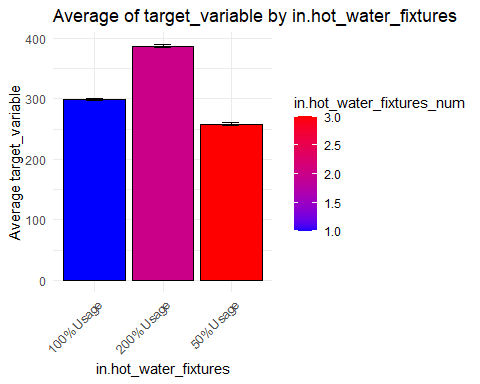
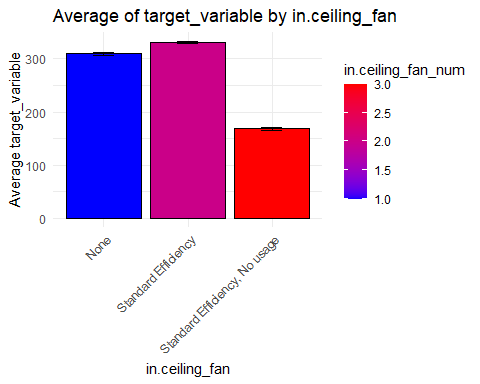
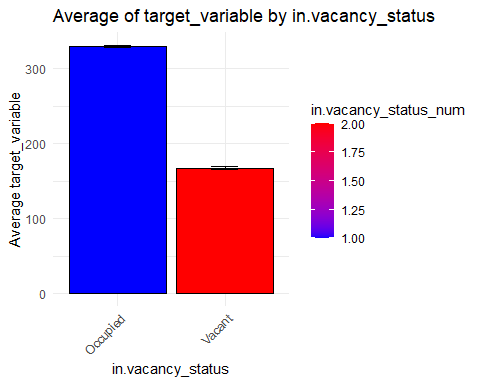
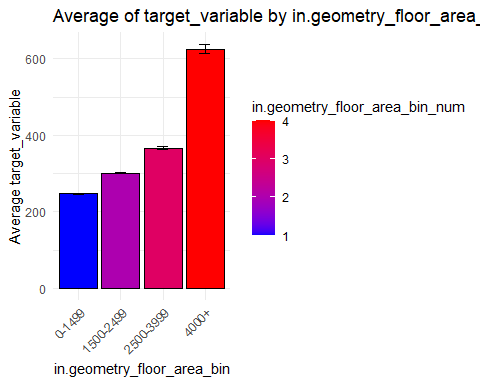
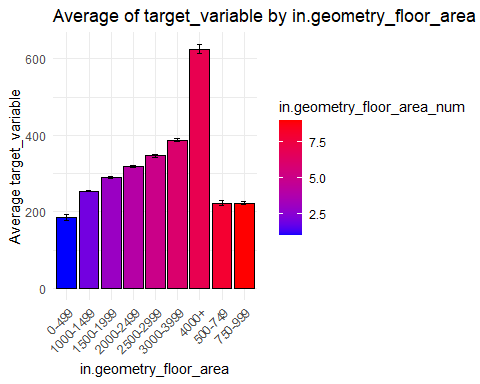
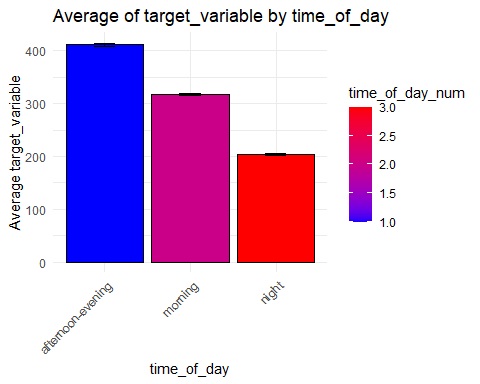
continuous\_variables <- c("`diffuse\_horizontal\_radiation\_[w/m2]`",  
 "`global\_horizontal\_radiation\_[w/m2]`",   
 "`dry\_bulb\_temperature\_[°c]`",   
 "`direct\_normal\_radiation\_[w/m2]`",   
 "`relative\_humidity\_[%]`",  
 "`wind\_speed\_[m/s]`",  
 "in.income")  
#   
continuous\_variables <- c("diffuse\_horizontal\_radiation\_[w/m2]",  
 "global\_horizontal\_radiation\_[w/m2]",  
 "dry\_bulb\_temperature\_[°c]",  
 "direct\_normal\_radiation\_[w/m2]",  
 "relative\_humidity\_[%]",  
 "wind\_speed\_[m/s]",  
 "in.income")  
#  
continuous\_data <- filtered\_data[, c("total\_energy\_consumption",  
 continuous\_variables)]  
#   
#   
# cor\_matrix <- cor(continuous\_data, use = "complete.obs")  
#   
# corrplot(cor\_matrix, method = "color")  
#   
#   
# corrplot(cor\_matrix, method = "color",  
# addCoef.col = "black", # Add correlation coefficients  
# tl.col = "black", # Color of text labels  
# tl.srt = 45, # Rotate text labels  
# cl.lim = c(-1, 1), # Set limits for color scale  
# col = colorRampPalette(c("blue", "white", "red"))(200))  
# continuous\_variables <- c("`diffuse\_horizontal\_radiation\_[w/m2]`",  
# "`global\_horizontal\_radiation\_[w/m2]`",   
# "`dry\_bulb\_temperature\_[°c]`",   
# "`direct\_normal\_radiation\_[w/m2]`",   
# "`relative\_humidity\_[%]`",  
# "`wind\_speed\_[m/s]`",  
# "in.income")

ggplot(filtered\_data, aes(x = time\_of\_day, y = total\_energy\_consumption)) +  
 geom\_line() +  
 labs(title = "Time Series of Total Energy Consumption",  
 x = "Time",  
 y = "Total Energy Consumption") +  
 theme\_minimal()



# install.packages("GGally")  
# library(GGally)  
#   
# # Select a subset of variables for the pair plot  
# selected\_data <- data[, continuous\_vars]  
#   
# ggpairs(continuous\_vars,  
# lower = list(continuous = "points", combo = "box"), # Custom lower panel  
# upper = list(continuous = "cor", combo = "facethist"), # Custom upper panel  
# diag = list(continuous = "densityDiag"), # Custom diagonal  
# progress = FALSE) +  
# theme\_light()

categorical\_variables <- c("time\_of\_day",  
 "in.geometry\_floor\_area",  
 "in.geometry\_floor\_area\_bin",  
 "in.vacancy\_status",  
 "in.ceiling\_fan",  
 "in.hot\_water\_fixtures",  
 "in.plug\_load\_diversity",  
 "in.usage\_level",  
 "upgrade.cooking\_range",  
 "in.clothes\_dryer",  
 "in.income",  
 "in.bedrooms",  
 "in.heating\_setpoint",  
 "in.cooling\_setpoint",  
 "in.cooking\_range")  
  
# Loop for bar plots with a continuous color scale  
for (cat\_var in categorical\_variables) {  
 # Convert the categorical variable to a numeric type for coloring  
 data[[paste0(cat\_var, "\_num")]] <- as.numeric(as.factor(data[[cat\_var]]))  
  
 plot <- ggplot(data, aes\_string(x = cat\_var, y = filtered\_data$total\_energy\_consumption, fill = paste0(cat\_var, "\_num"))) +  
 geom\_bar(stat = "summary", fun = "mean", position = position\_dodge(), color = "black") +  
 geom\_errorbar(stat = "summary", fun.data = "mean\_se", fun.args = list(mult = 1), position = position\_dodge(0.9), width = 0.25) +  
 scale\_fill\_gradient(low = "blue", high = "red") + # Continuous color scale  
 labs(title = paste("Average of", "target\_variable", "by", cat\_var),  
 x = cat\_var,  
 y = "Average target\_variable") +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) # Rotate x-axis labels for readability  
  
 # Remove the temporary numeric column  
 data[[paste0(cat\_var, "\_num")]] <- NULL  
  
 print(plot)  
}



#install.packages("plotly")  
# library(plotly)  
#   
# plot\_ly(filtered\_data, x = ~`dry\_bulb\_temperature\_[°c]`, y = ~total\_energy\_consumption, z = ~`time\_of\_day`,  
# type = 'scatter3d', mode = 'markers',  
# marker = list(size = 2, color = ~`time\_of\_day`, colorscale = 'Viridis')) %>%  
# layout(title = '3D Scatter Plot',  
# scene = list(xaxis = list(title = 'Dry Bulb Temperature [°C]'),  
# yaxis = list(title = 'Total Energy Consumption'),  
# zaxis = list(title = 'Time of Day')))  
  
continuous\_variables <- c("`diffuse\_horizontal\_radiation\_[w/m2]`",  
 "`global\_horizontal\_radiation\_[w/m2]`",   
 "`dry\_bulb\_temperature\_[°c]`",   
 "`direct\_normal\_radiation\_[w/m2]`",   
 "`relative\_humidity\_[%]`",  
 "`wind\_speed\_[m/s]`",  
 "in.income")  
  
for (var in continuous\_variables) {  
 plot <- ggplot(filtered\_data, aes\_string(x = var, y = filtered\_data$total\_energy\_consumption, color = "time\_of\_day")) +  
 geom\_point() +  
 labs(title = paste("Total Energy Consumption vs", var, "by Time of Day"),  
 x = var,  
 y = "Total Energy Consumption") +  
 scale\_color\_brewer(palette = "Set1") +  
 theme\_minimal() +  
 theme(legend.position = "right")  
   
 print(plot)  
}

