Project 1

Varsha Rajesh, Elizabeth Szwajnos, Neha Sancheti

Overview

In the `data` directory of this project you will find the file from a paper published in *Nature Energy* titled Natural gas savings in Germany during the 2022 energy crisis. Here is the abstract of the article:

> Russia curbed its natural gas supply to Europe in 2021 and 2022, creating a grave energy crisis. This Article empirically estimates the crisis response of natural gas consumers in Germany—for decades, the largest export market for Russian gas. Using a multiple regression model, we estimate the response of small consumers, industry and power stations separately, controlling for the nonlinear temperature-heating relationship, seasonality and trends. We find significant and substantial gas savings for all consumer groups, but with differences in timing and size. For instance, industry started reducing consumption as early as September 2021, while small consumers saved substantially only since March 2022. Across all sectors, gas consumption during the second half of 2022 was 23% below the temperature-adjusted baseline. We discuss the drivers behind these savings and draw conclusions on their role in coping with the crisis.

Your job in this project falls into two categories:

1. A set of **tasks** that your group must complete exactly
2. A set of **objectives** that are more general in their approach.

Tasks

Task 1

- Load two files. To work in the console, use the `Session -> Set Working Directory -> To Source File Location`.
  - Call the first table `daily`: "./data/natural_gas_germany_daily.csv"
  - Call the second table `gas`: "./data/dutch_ttf_natural_gas.csv". Be sure to properly import the `Date` column.
  - Demonstrate that these have been loaded by showing the number of rows and columns in each table.

```
library(tidyverse)
library(lubridate)

# Load the datasets
daily <- read_csv("./data/natural_gas_germany_daily.csv", col_types = cols(date = col_guess()))
gas <- read_csv("./data/dutch_ttf_natural_gas.csv", col_types = cols(Date = col_guess()))

if (!is.character(gas$Date)) {
  gas$Date <- as.character(gas$Date)
}
#properly importing data col for gas
gas$Date <- parse_date(gas$Date, "%m/%d/%Y", locale = locale("en"))
```

```
dim_daily <- dim(daily)
dim_gas <- dim(gas)

print(paste("Daily Data Dimensions:", dim_daily[1], "rows and", dim_daily[2], "columns"))
```

```
## [1] "Daily Data Dimensions: 2191 rows and 19 columns"
```

```
print(paste("Gas Data Dimensions after cleaning:", dim_gas[1], "rows and", dim_gas[2], "columns"))
```

```
## [1] "Gas Data Dimensions after cleaning: 1346 rows and 7 columns"
```

Task 2

- The data in `daily` are collected over days, with information on different types of natural gas consumption (`consumption_small`, `consumption_industry`, `consumption_power`). Provide summaries of typical values for each of these three types of consumption.

```
daily |>
  summarize(
    small_mean = mean(consumption_small, na.rm = TRUE),
    small_median = median(consumption_small, na.rm = TRUE),
    small_sd = sd(consumption_small, na.rm = TRUE),
    small_min = min(consumption_small, na.rm = TRUE),
    small_max = max(consumption_small, na.rm = TRUE),
    small_iqr = IQR(consumption_small, na.rm = TRUE),

    industry_mean = mean(consumption_industry, na.rm = TRUE),
    industry_median = median(consumption_industry, na.rm = TRUE),
    industry_sd = sd(consumption_industry, na.rm = TRUE),
    industry_min = min(consumption_industry, na.rm = TRUE),
    industry_max = max(consumption_industry, na.rm = TRUE),
    industry_iqr = IQR(consumption_industry, na.rm = TRUE),

    power_mean = mean(consumption_power, na.rm = TRUE),
    power_median = median(consumption_power, na.rm = TRUE),
    power_sd = sd(consumption_power, na.rm = TRUE),
    power_min = min(consumption_power, na.rm = TRUE),
    power_max = max(consumption_power, na.rm = TRUE),
    power_iqr = IQR(consumption_power, na.rm = TRUE)
  )
```

```
## # A tibble: 1 x 18
##   small_mean small_median small_sd small_min small_max small_iqr industry_mean
##        <dbl>        <dbl>    <dbl>     <dbl>     <dbl>     <dbl>         <dbl>
## 1       1.07        0.899    0.759     0.163      3.28      1.44          1.20
## # i 11 more variables: industry_median <dbl>, industry_sd <dbl>,
## #   industry_min <dbl>, industry_max <dbl>, industry_iqr <dbl>,
## #   power_mean <dbl>, power_median <dbl>, power_sd <dbl>, power_min <dbl>,
## #   power_max <dbl>, power_iqr <dbl>
```

Task 3

Answer some questions about the data in `daily`:

- How many weeks do the data cover?

- What is the percentage change in the `consumption_*` variables (that is the last day minus the first day divided by the first day)?
- What proportion of the days are marked as holidays?
- For each month in each year, what was the year-month combination with the lowest median `consumption_power` value?

```
# calculate number of weeks
num_weeks <- as.numeric(difftime(max(daily$date, na.rm = TRUE),
                                 min(daily$date, na.rm = TRUE),
                                 units = "weeks"))
print(paste("The data cover", num_weeks, "weeks."))
```

```
## [1] "The data cover 312.857142857143 weeks."
```

```
percentage_change <- daily |>
  summarize(
    small_change = 100 * (last(consumption_small) - first(consumption_small)) / first(consumption_small)
    industry_change = 100 * (last(consumption_industry) - first(consumption_industry)) / first(consumpti
    power_change = 100 * (last(consumption_power) - first(consumption_power)) / first(consumption_power)
  )

print(percentage_change)
```

```
## # A tibble: 1 x 3
##   small_change industry_change power_change
##          <dbl>           <dbl>        <dbl>
## 1           NA           -49.5        -62.0
```

```
holiday_prop <- mean(daily$holiday == 1, na.rm = TRUE)
print(paste("Proportion of holidays:", holiday_prop))
```

```
## [1] "Proportion of holidays: 0.0360565951620265"
```

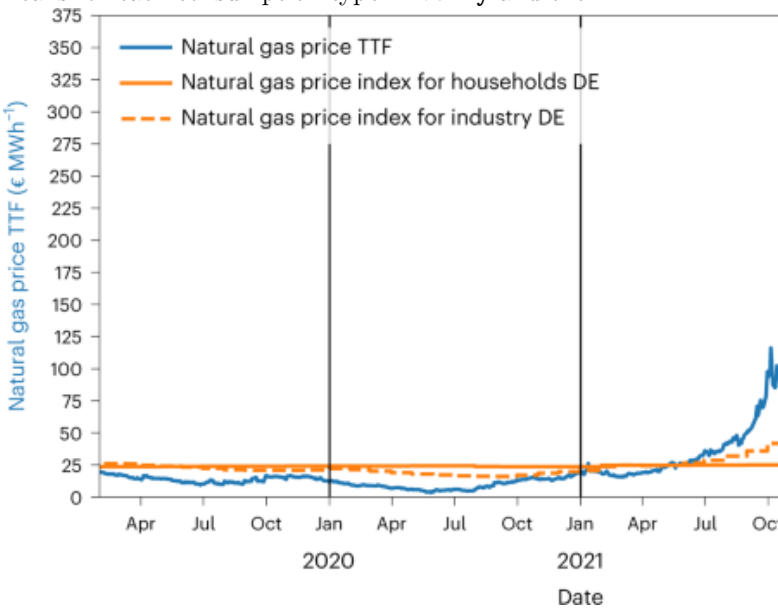
```
daily$date <- as.Date(daily$date, format="%Y-%m-%d")

daily$year_month <- format(daily$date, "%Y-%m")

median_consumption <- aggregate(consumption_power ~ year_month, data = daily, median, na.rm = TRUE)

lowest_median <- median_consumption[which.min(median_consumption$consumption_power), ]

lowest_median
```

```
##    year_month consumption_power
## 56    2021-08         0.1634914
```

This data covers almost 313 full weeks.

Task 4

- The original paper aggregated the data to monthly means for each consumption type in `daily` and the `Price` column of `gas` to produce the following image:



  Produce plots that show the same information that is presented in this plot. Your plots do not have to have the same colors or markings, but we should be able to use them to compare the trends for the three price variables.

```
library(tidyverse)
library(lubridate)

daily <- daily |>
  mutate(date = as.Date(date))

gas <- gas |>
  mutate(Date = as.Date(Date, format="%Y-%m-%d"))

# zggregate
daily_monthly <- daily |>
  mutate(year_month = floor_date(date, "month")) |>
  group_by(year_month) |>
  summarize(
    consumption_small = mean(consumption_small, na.rm = TRUE),
    consumption_industry = mean(consumption_industry, na.rm = TRUE),
    consumption_power = mean(consumption_power, na.rm = TRUE)
  )

gas_monthly <- gas |>
  mutate(year_month = floor_date(Date, "month")) |>
  group_by(year_month) |>
  summarize(Price = mean(Price, na.rm = TRUE))

# Merge datasets
monthly_data <- left_join(daily_monthly, gas_monthly, by = "year_month")

#plot
ggplot(monthly_data, aes(x = year_month)) +
```

```
  geom_line(aes(y = Price, color = "Natural gas price TTF (Euro / MWh)"), size = 1) +
  geom_line(aes(y = consumption_small * 10, color = "Natural gas price index for households DE"),
            size = 1, linetype = "solid") +
  geom_line(aes(y = consumption_industry * 10, color = "Natural gas price index for industry DE"),
            size = 1, linetype = "dashed") +
  scale_y_continuous(
    name = "Natural gas price TTF (€ / MWh)",
    sec.axis = sec_axis(~ . / 10, name = "Natural gas price indices (2015 = 100)")
  ) +
  labs(title = "Natural Gas Prices and Price Indices in Germany (Monthly,2020- 2022)",
       x = "Date",
       color = "Legend") +
  scale_color_manual(values = c("blue", "orange", "red")) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        axis.title = element_text(size = 12),
        legend.title = element_text(size = 10),
        legend.text = element_text(size = 9))
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: Removed 9 rows containing missing values (`geom_line()`).
```

```
## Warning: Removed 12 rows containing missing values (`geom_line()`).
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Natural gas price TTF (€ / MWh)' in 'mbcsToSbcs': dot
## substituted for <e2>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Natural gas price TTF (€ / MWh)' in 'mbcsToSbcs': dot
## substituted for <82>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Natural gas price TTF (€ / MWh)' in 'mbcsToSbcs': dot
## substituted for <ac>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Natural gas price TTF (€ / MWh)' in 'mbcsToSbcs': dot
## substituted for <e2>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Natural gas price TTF (€ / MWh)' in 'mbcsToSbcs': dot
## substituted for <82>
```

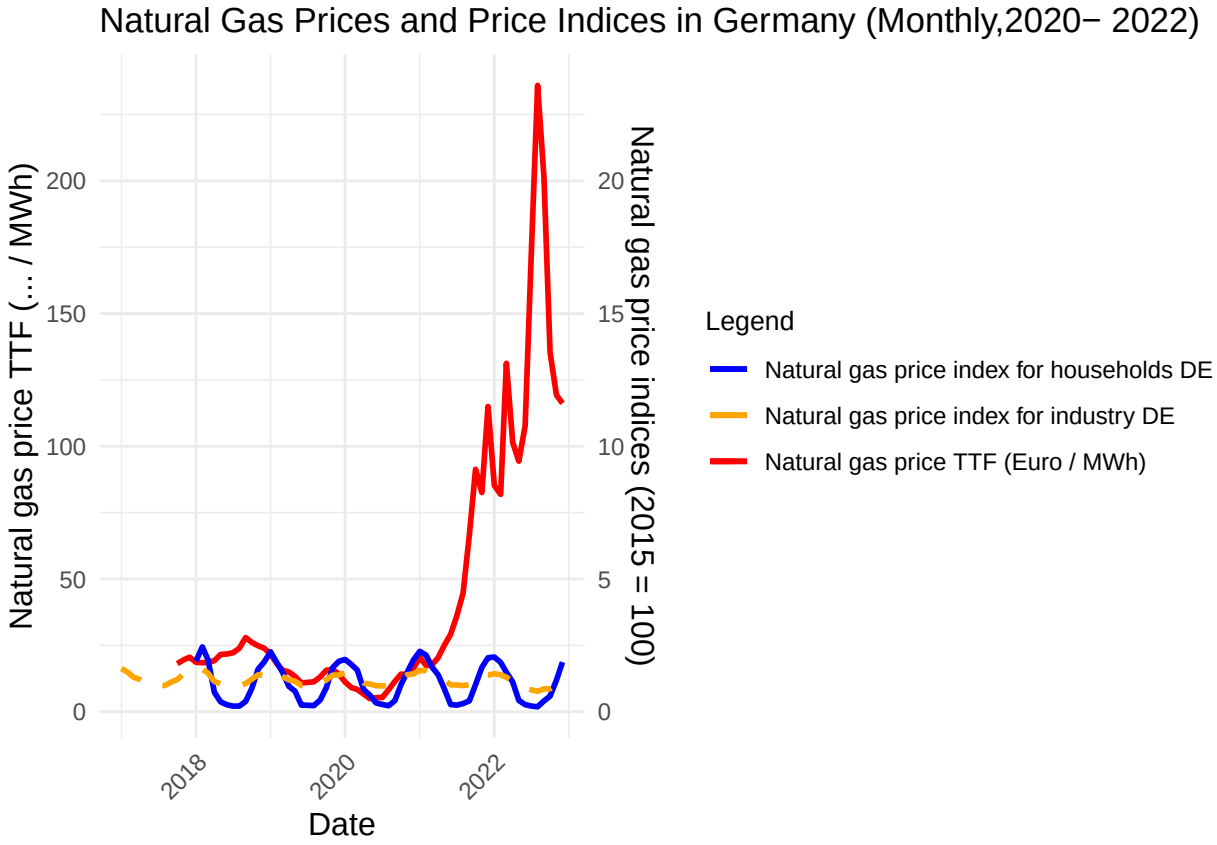
```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Natural gas price TTF (€ / MWh)' in 'mbcsToSbcs': dot
## substituted for <ac>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Natural gas price TTF (€ / MWh)' in 'mbcsToSbcs': dot
## substituted for <e2>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Natural gas price TTF (€ / MWh)' in 'mbcsToSbcs': dot
```

```
## substituted for <82>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Natural gas price TTF (€ / MWh)' in 'mbcsToSbcs': dot
## substituted for <ac>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Natural gas price TTF (€ / MWh)' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Natural gas price TTF (€ / MWh)' in 'mbcsToSbcs': dot
## substituted for <82>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Natural gas price TTF (€ / MWh)' in 'mbcsToSbcs': dot
## substituted for <ac>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Natural gas price TTF (€ / MWh)' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Natural gas price TTF (€ / MWh)' in 'mbcsToSbcs': dot
## substituted for <82>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Natural gas price TTF (€ / MWh)' in 'mbcsToSbcs': dot
## substituted for <ac>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Natural gas price TTF (€ / MWh)' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Natural gas price TTF (€ / MWh)' in 'mbcsToSbcs': dot
## substituted for <82>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Natural gas price TTF (€ / MWh)' in 'mbcsToSbcs': dot
## substituted for <ac>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Natural gas price TTF (€ / MWh)' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Natural gas price TTF (€ / MWh)' in 'mbcsToSbcs': dot
## substituted for <82>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Natural gas price TTF (€ / MWh)' in 'mbcsToSbcs': dot
## substituted for <ac>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Natural gas price TTF (€ / MWh)' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Natural gas price TTF (€ / MWh)' in 'mbcsToSbcs': dot
## substituted for <82>
```

```
## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Natural gas price TTF (€ / MWh)' in 'mbcsToSbcs': dot
## substituted for <ac>
```



Task 5

- Write a predicate function that returns true if any value in vector is missing. Use this function to find columns with missing values in the `daily` column. Create a plot or table that shows how often patterns of missingness occur: are all of the missing values in the same rows or are the various columns missing data in different ways?

```
#predicate function to check for NA values
has_missing_values <- function(x) {
  any(is.na(x))
}

# find columns with missing values in `daily`
missing_columns <- names(daily)[sapply(daily, has_missing_values)]

# check patterns of missingness
missing_patterns <- daily %>%
  mutate(missing_pattern = apply(., 1, function(row) paste(ifelse(is.na(row), 'NA', '.'), collapse = ''
  group_by(missing_pattern) %>%
  summarise(count = n()) %>%
  arrange(desc(count))

# Show missing patterns
missing_patterns
```

```
## # A tibble: 4 x 2
##   missing_pattern          count
##   <chr>                    <int>
## 1 .....................     1825
## 2 .NA.......NANA.........    276
## 3 .NA......NANANA.........    89
## 4 ..NANA................       1
```

Task 6

- Limit the gas table to days where the price exceeded the yearly median. Use the concept of circular means to compute the average day of the year when price exceeds the yearly median price. The yday function will likely be useful here.

```
median_price <- median(gas$Price, na.rm = TRUE)

filtered_gas <- gas |>
  filter(Price > median_price) |>
  mutate(yday = yday(Date))

angles <- filtered_gas$yday * 2 * pi / 365

mean_x <- mean(cos(angles), na.rm = TRUE)
mean_y <- mean(sin(angles), na.rm = TRUE)

circular_mean_angle <- atan2(mean_y, mean_x)

circular_mean_day <- (circular_mean_angle / (2 * pi)) * 365
circular_mean_day <- ifelse(circular_mean_day < 0, circular_mean_day + 365, circular_mean_day)

cat("Average day of the year when price exceeds the yearly median price:", round(circular_mean_day), "\n
```

```
## Average day of the year when price exceeds the yearly median price: 272
```

Task 7

- Using the cut function, create two nominal variables from quantitative data in the daily dataset. Use these groups to summarize the data. Use arrange to show the smallest or largest values in these comparisons.

```
daily <- daily |>
  mutate(date = as.Date(date))

# nominal DayType variable (Weekday or Weekend)
daily <- daily |>
  mutate(DayType = ifelse(weekdays(date) %in% c("Saturday", "Sunday"), "Weekend", "Weekday"))

# nominal MonthCategory variable (Season)
daily <- daily |>
  mutate(MonthCategory = case_when(
    as.numeric(format(date, "%m")) %in% c(12, 1, 2) ~ "Winter",
    as.numeric(format(date, "%m")) %in% c(3, 4, 5) ~ "Spring",
    as.numeric(format(date, "%m")) %in% c(6, 7, 8) ~ "Summer",
    as.numeric(format(date, "%m")) %in% c(9, 10, 11) ~ "Fall",
    TRUE ~ "Unknown"  # Fallback in case of unexpected values
  ))
```

```
data_summary <- daily |>
  group_by(DayType, MonthCategory) |>
  summarize(
    MeanConsumption = mean(consumption_small, na.rm = TRUE),
    MedianConsumption = median(consumption_small, na.rm = TRUE),
    TotalDays = n(),
    .groups = "drop"
  )


data_summary |>
  arrange(MeanConsumption)
```

```
## # A tibble: 8 x 5
##   DayType MonthCategory MeanConsumption MedianConsumption TotalDays
##   <chr>   <chr>                   <dbl>             <dbl>     <int>
## 1 Weekend Summer                  0.242             0.228       157
## 2 Weekday Summer                  0.250             0.241       395
## 3 Weekend Fall                    0.921             0.810       156
## 4 Weekday Fall                    0.944             0.835       390
## 5 Weekend Spring                  1.08              0.956       157
## 6 Weekday Spring                  1.09              0.978       395
## 7 Weekend Winter                  1.99              1.98        156
## 8 Weekday Winter                  2.02              2.04        385
```

Task 8

- There are several variables that pull out data by different industry (the `_idx` columns). Create a table for these columns using `select` and the `ends_with` function. Provide two different plots that show of the relations between these variables (you do not need to have all variables in each plot).

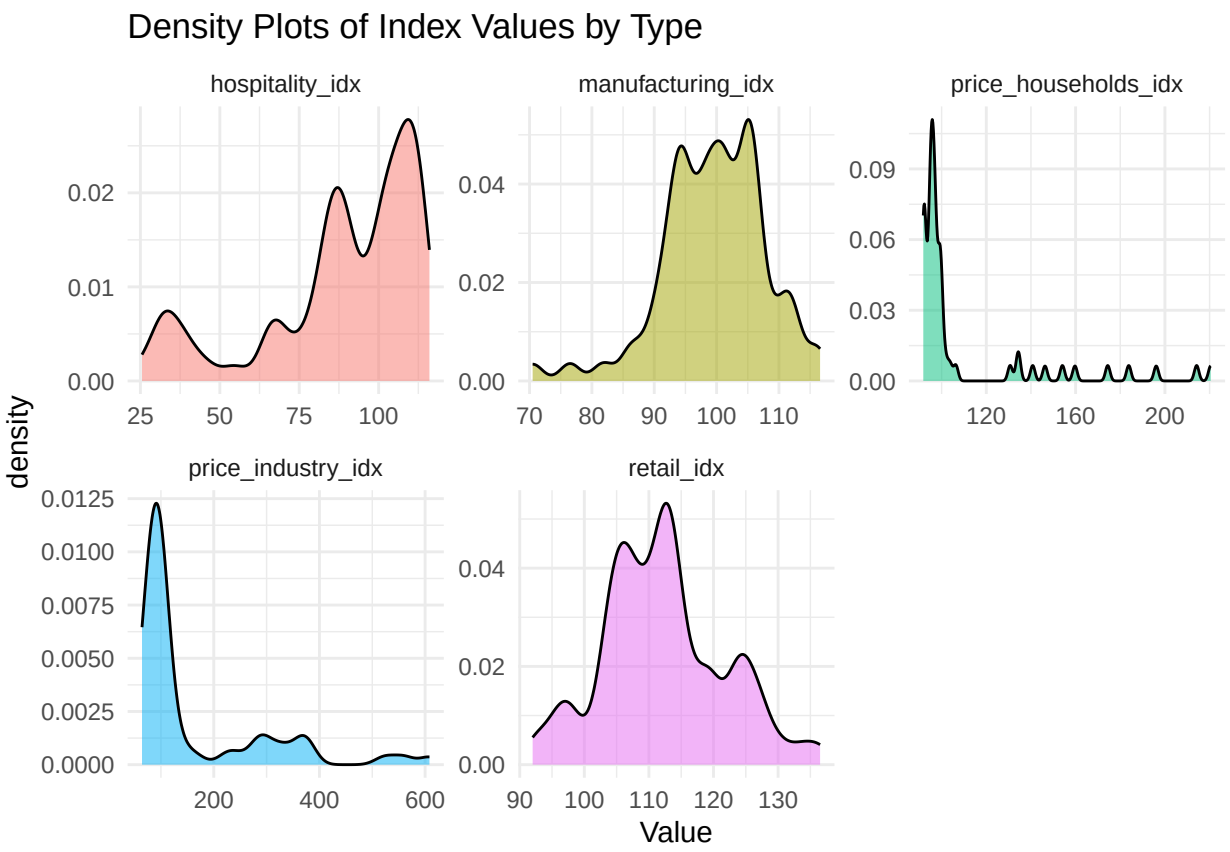
```
# Select columns that end with '_idx'
industry_data <- daily |>
  select(ends_with("_idx"))

head(industry_data)
```

```
## # A tibble: 6 x 5
##   manufacturing_idx hospitality_idx retail_idx price_industry_idx
##               <dbl>           <dbl>      <dbl>              <dbl>
## 1              94.1            82.6       96.1                 NA
## 2              94.1            82.6       96.1                 NA
## 3              94.1            82.6       96.1                 NA
## 4              94.1            82.6       96.1                 NA
## 5              94.1            82.6       96.1                 NA
## 6              94.1            82.6       96.1                 NA
## # i 1 more variable: price_households_idx <dbl>
```

```
industry_long <- industry_data |>
  pivot_longer(cols = everything(), names_to = "Index", values_to = "Value")

box_plot <- ggplot(industry_long, aes(x = Index, y = Value)) +
  geom_boxplot(aes(fill = Index), outlier.alpha = 0.3) +
  labs(title = "Box Plot of Industry and Price Indices",
```

```
       x = "Index Type",
       y = "Index Value") +
  theme_minimal() +
  theme(legend.position = "none")

# Density Plot for Each Index
density_plot <- ggplot(industry_long, aes(x = Value, fill = Index)) +
  geom_density(alpha = 0.5) +
  facet_wrap(~ Index, scales = "free") +
  labs(title = "Density Plots of Index Values by Type") +
  theme_minimal() +
  theme(legend.position = "none")

print(density_plot)
```

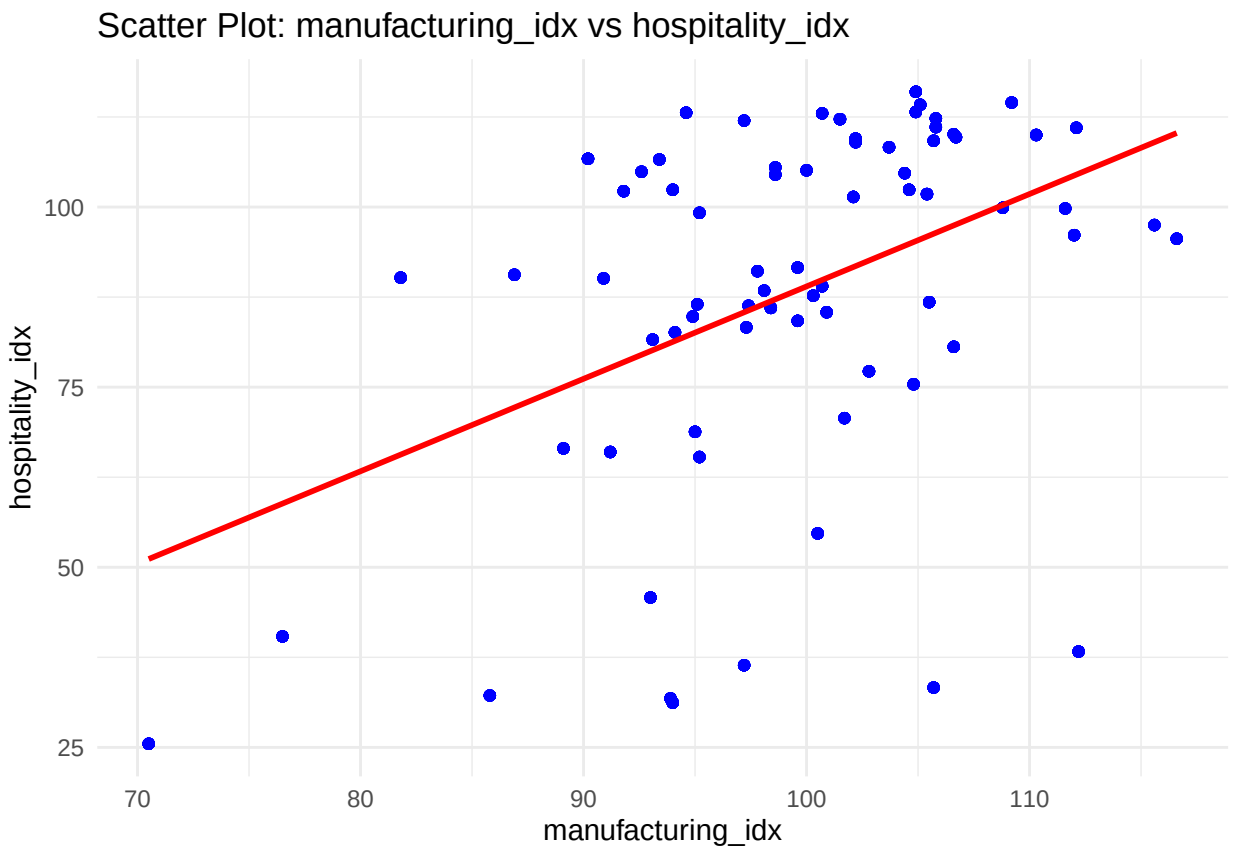
```
## Warning: Removed 730 rows containing non-finite values (`stat_density()`).
```




```
library(ggplot2)


idx1 <- names(industry_data)[1]
idx2 <- names(industry_data)[2]

# Scatter plot between two index variables
scatter_plot <- ggplot(daily, aes(x = .data[[idx1]], y = .data[[idx2]])) +
  geom_point(alpha = 0.5, color = "blue") +
  geom_smooth(method = "lm", color = "red", se = FALSE) +
```

```
  labs(title = paste("Scatter Plot:", idx1, "vs", idx2),
       x = idx1, y = idx2) +
  theme_minimal()

print(scatter_plot)
```

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



The manufacturing_idx and hospitality_idx have positive correlation, which means economic activity likely affects their gas consumption in the two industries similarly. We can see that the density of hospitality and manufacturing are left skewed while price_households and price_industry are heavily right skewed. retail_idx is largely symmetric.

Objectives

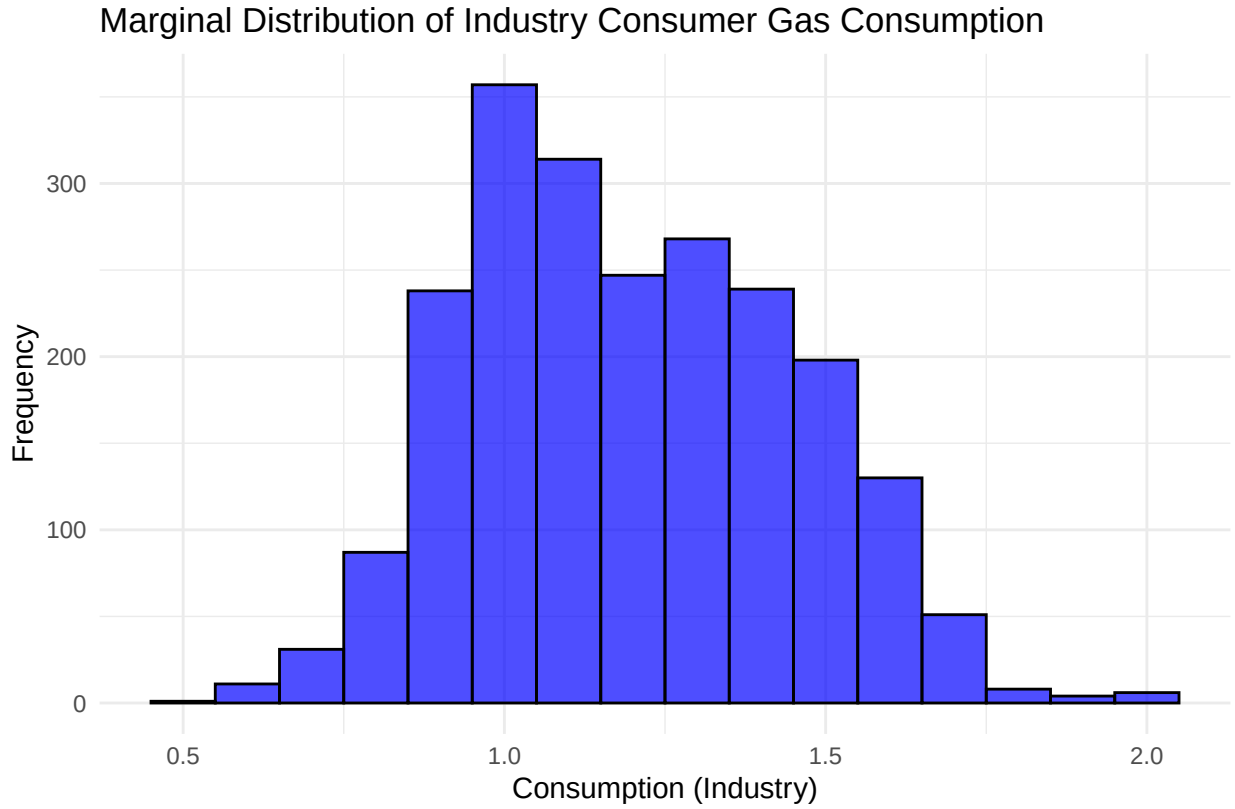
Objective 1

- Produce at least five more figures. For each figure, write a brief caption explaining the plot and what you have learned from the plot. Each figure should attempt to provide new insight into the data set not included elsewhere
  - A marginal distribution
  - A joint distribution
  - A plot of a summary measure such as a conditional mean
  - A plot using `facet_wrap` or `facet_grid`
  - A plot that shows seasonal effects before the crisis (September 2021 until October 2022)

```
# 1: Marginal Distribution of Industry Consumer Gas consumption
ggplot(daily, aes(x = consumption_industry)) +
  geom_histogram(binwidth = 0.1, fill = "blue", color = "black", alpha = 0.7) +
```

```
  labs(title = "Marginal Distribution of Industry Consumer Gas Consumption", x = "Consumption (Industry)
  theme_minimal()
```

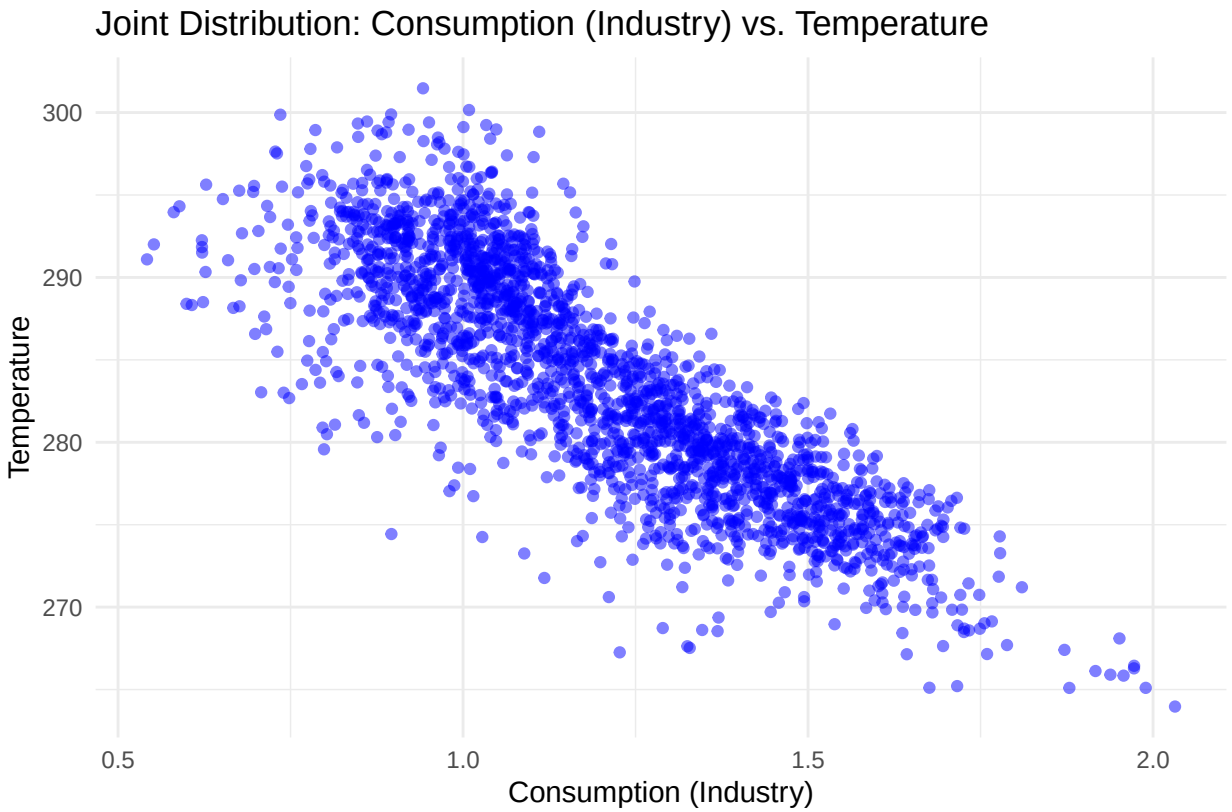
```
## Warning: Removed 1 rows containing non-finite values (`stat_bin()`).
```



l to adapt quickly, showing the large shift in usages. This aligns with how the industries led the response to the crisis.

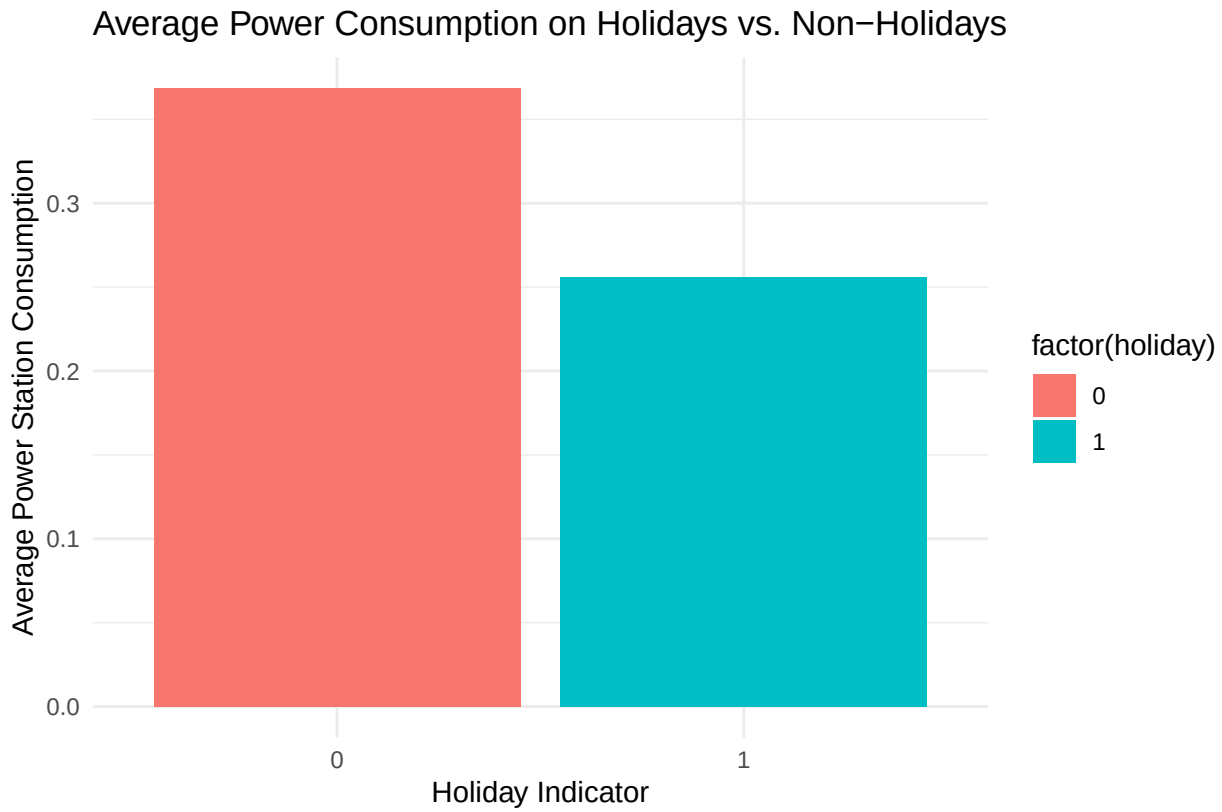
```
#2: Joint Distribution of consumption_industry and temperature
ggplot(daily, aes(x = consumption_industry, y = temperature)) +
  geom_point(alpha = 0.5, color = "blue") +
  labs(title = "Joint Distribution: Consumption (Industry) vs. Temperature",
       x = "Consumption (Industry)",
       y = "Temperature", caption = "scatter plot showing negative correlation between industrial natura
  theme_minimal()
```

```
## Warning: Removed 1 rows containing missing values (`geom_point()`).
```



ıich highlights how much seasons affect industrial energy demand during the crisis. This may be because of heating.

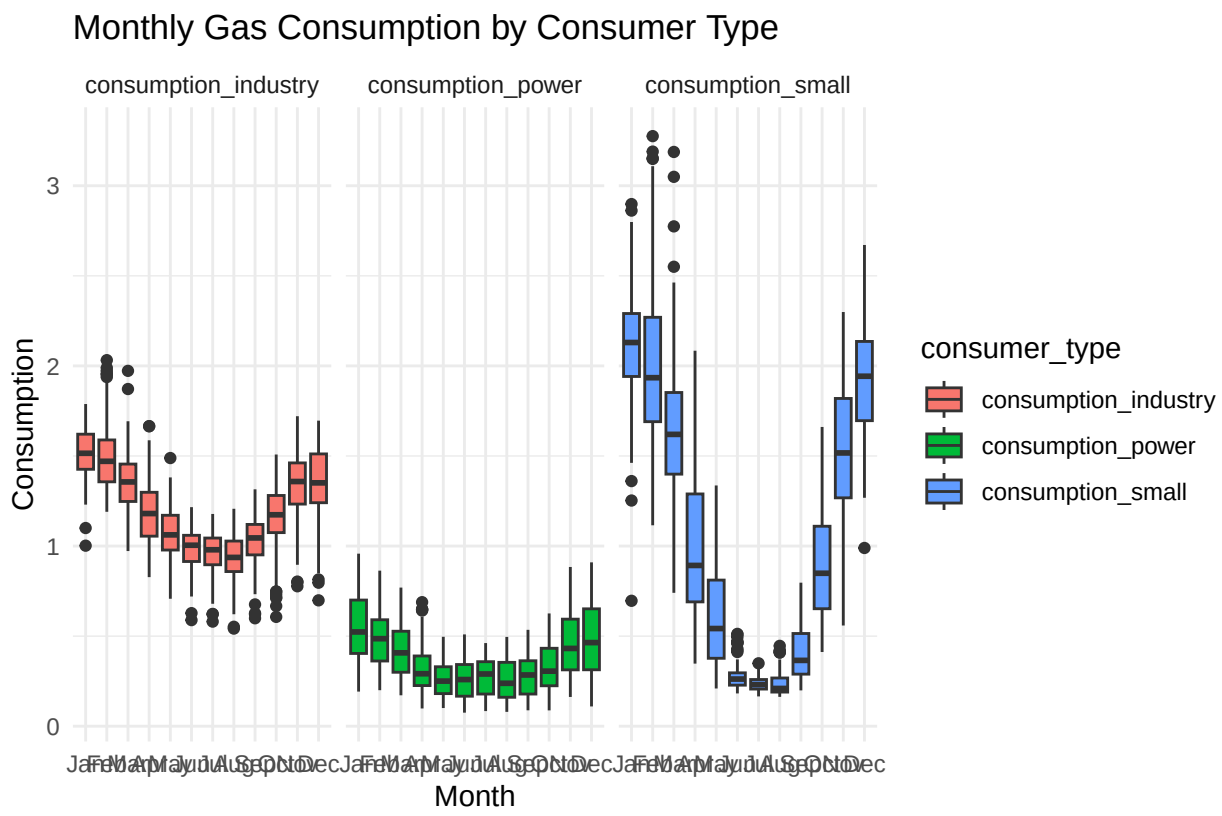
```
#3: Conditional Mean Plot for average consumption power grouped by if holiday or not
daily |>
  group_by(holiday) |>
  summarize(mean_consumption_power = mean(consumption_power, na.rm = TRUE)) |>
  ggplot(aes(x = factor(holiday), y = mean_consumption_power, fill = factor(holiday))) +
  geom_bar(stat = "identity") +
  labs(title = "Average Power Consumption on Holidays vs. Non-Holidays", x = "Holiday Indicator", y = "
  theme_minimal()
```



must be not producing much on holidays because less energy is needed, so demand is lower.

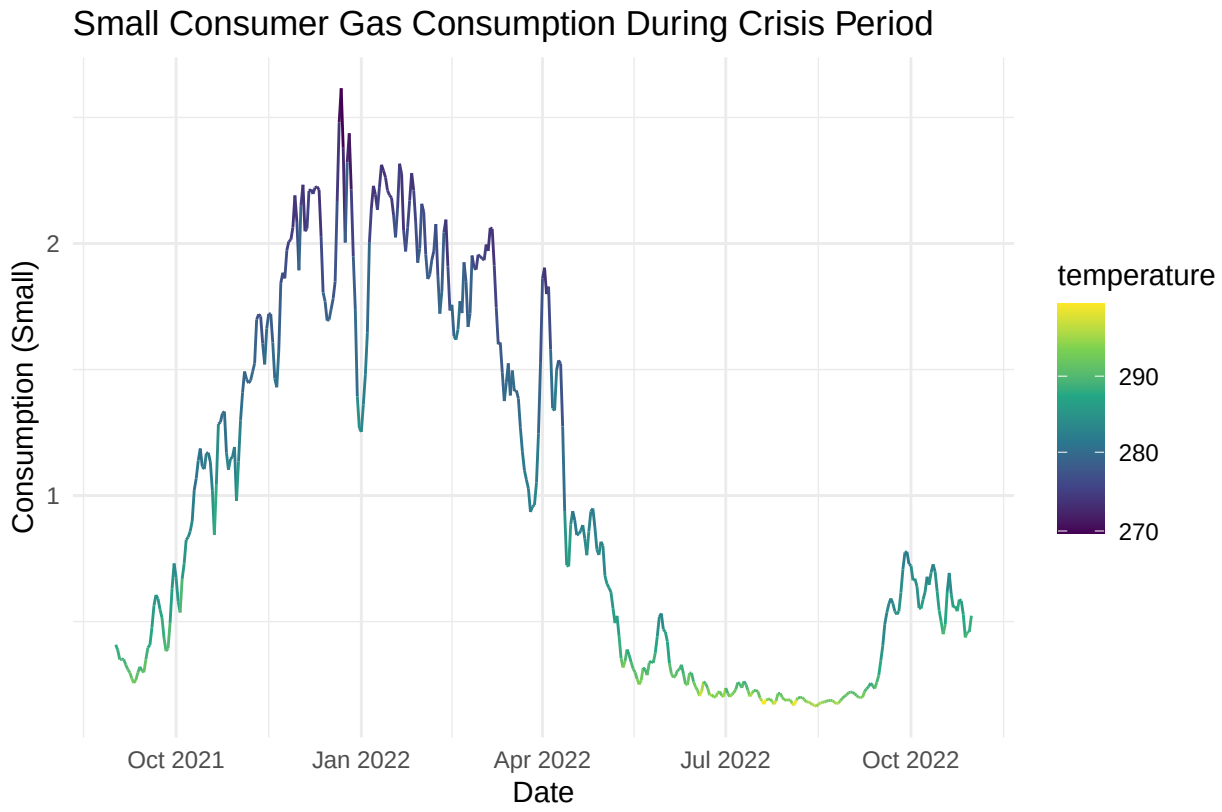
```
#4: Facet Plot Consumption by month and type
daily |>
  mutate(month = month(date, label = TRUE)) |>
  pivot_longer(cols = starts_with("consumption_"), names_to = "consumer_type", values_to = "consumption
  ggplot(aes(x = month, y = consumption, fill = consumer_type)) +
  geom_boxplot() +
  facet_wrap(~ consumer_type) +
  labs(title = "Monthly Gas Consumption by Consumer Type", x = "Month", y = "Consumption", caption = "M
  theme_minimal()
```

```
## Warning: Removed 367 rows containing non-finite values (`stat_boxplot()`).
```



ıll has highest variability. For consumption_industry, there is a more moderate level.

```
#5: Seasonal Effects Before Crisis
daily |>
  filter(date >= as.Date("2021-09-01") & date <= as.Date("2022-10-31")) |>
  ggplot(aes(x = date, y = consumption_small, color = temperature)) +
  geom_line() +
  labs(title = "Small Consumer Gas Consumption During Crisis Period", x = "Date", y = "Consumption (Sma
  theme_minimal() +
  scale_color_viridis_c(trans = "log")
```



s's impact on small consumer behavior. It then decreases drastically in the second half of 2022.

Objective 2

- Compare and contrast holidays and non-holidays for household energy consumption. Select 3 ways of comparing these groups. Provide at least one graph.

```
median_holiday <- median(daily$consumption_small[daily$holiday == 1], na.rm = TRUE)
median_non_holiday <- median(daily$consumption_small[daily$holiday == 0], na.rm = TRUE)

print(median_holiday)
```

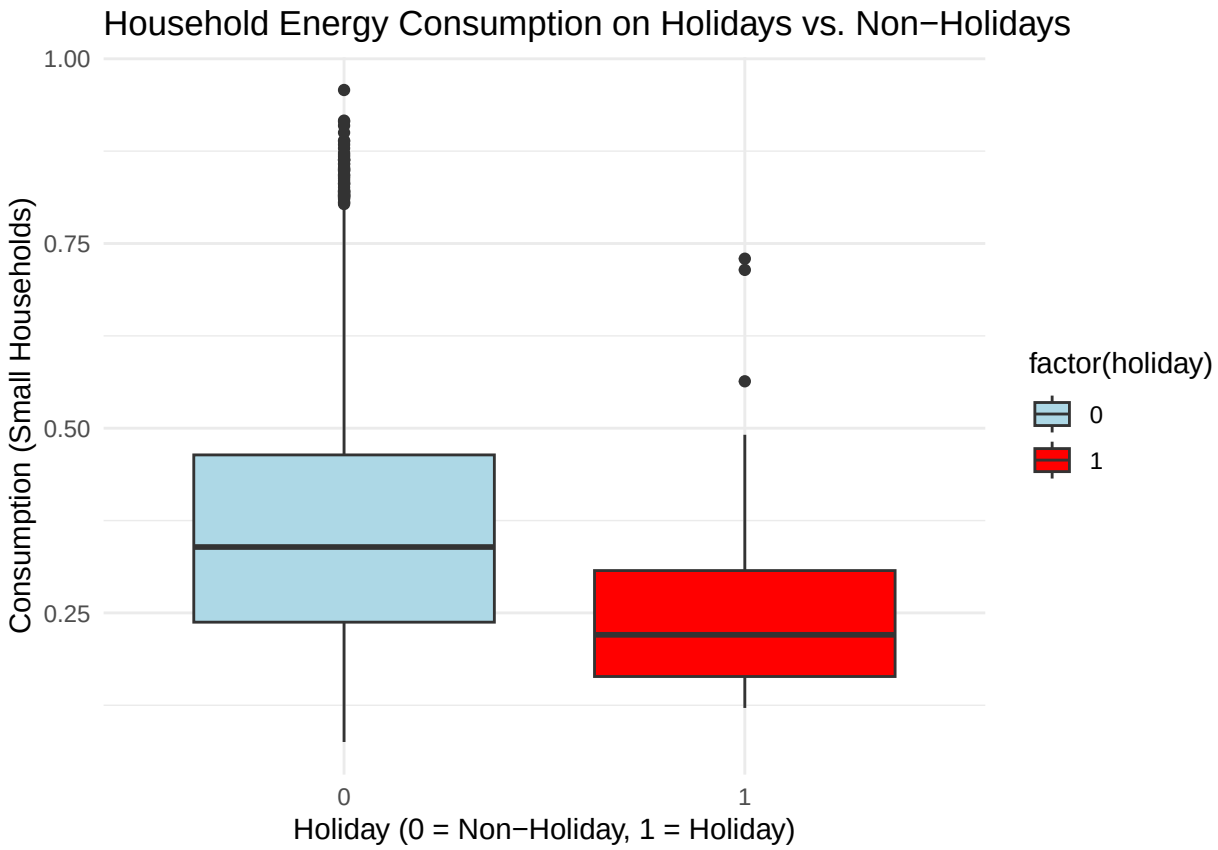
```
## [1] 0.8432533
```

```
print(median_non_holiday)
```

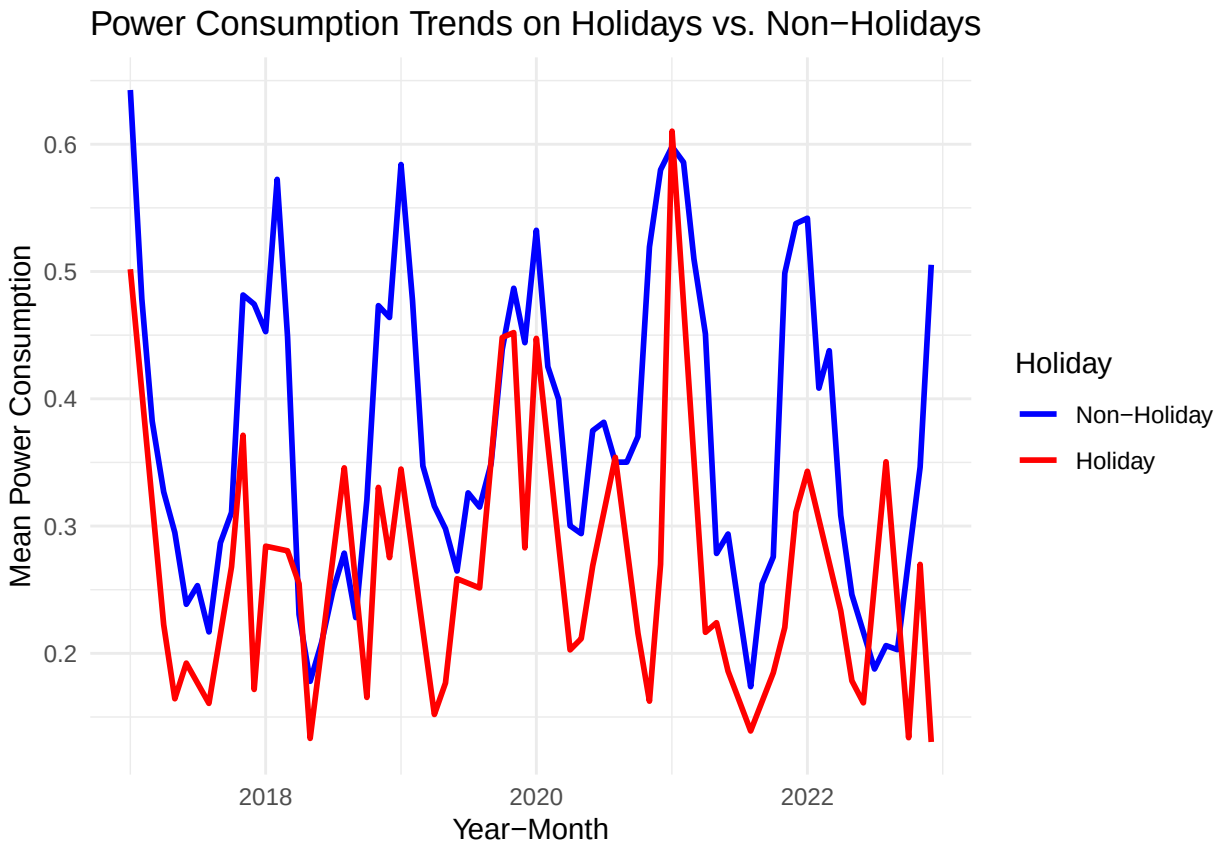
```
## [1] 0.901598
```

```
# Create a boxplot to visualize distribution
ggplot(daily, aes(x = factor(holiday), y = consumption_power, fill = factor(holiday))) +
  geom_boxplot() +
  labs(title = "Household Energy Consumption on Holidays vs. Non-Holidays",
       x = "Holiday (0 = Non-Holiday, 1 = Holiday)",
       y = "Consumption (Small Households)") +
  scale_fill_manual(values = c("lightblue", "red")) +
  theme_minimal()
```

```
## Warning: Removed 1 rows containing non-finite values (`stat_boxplot()`).
```




```r
daily$date <- as.Date(daily$date)
# Create year-month column
daily$year_month <- format(daily$date, "%Y-%m")

# Compute mean consumption per year-month for holidays and non-holidays
trend_data <- aggregate(consumption_power ~ year_month + holiday, data = daily, mean, na.rm = TRUE)

# Convert year-month to Date format (1st of the month)
trend_data$year_month <- as.Date(paste0(trend_data$year_month, "-01"))

# Plot trend over time
ggplot(trend_data, aes(x = year_month, y = consumption_power, color = factor(holiday))) +
  geom_line(linewidth = 1) +
  labs(title = "Power Consumption Trends on Holidays vs. Non-Holidays",
       x = "Year-Month",
       y = "Mean Power Consumption",
       color = "Holiday") +
  scale_color_manual(values = c("blue", "red"), labels = c("Non-Holiday", "Holiday")) +
  theme_minimal()
```




```
#find stats for holiday or non-holiday
holiday_comparison <- daily |>
  group_by(holiday) |>
  summarize(
    MeanConsumption = mean(consumption_small, na.rm = TRUE),
    MedianConsumption = median(consumption_small, na.rm = TRUE),
    SDConsumption = sd(consumption_small, na.rm = TRUE),
    TotalDays = n(),
    .groups = "drop"
  )

print(holiday_comparison)
```

```
## # A tibble: 2 x 5
##   holiday MeanConsumption MedianConsumption SDConsumption TotalDays
##     <dbl>           <dbl>             <dbl>         <dbl>     <int>
## 1       0            1.07             0.902         0.762      2112
## 2       1            1.04             0.843         0.685        79
```

We used three different ways to compare and contrast the holidays and non-holidays for household energy consumption. The first method used was comparing the medians for the consumption power for when it was a holiday vs. when it wasn't a holiday. It can be seen that the median consumption power was a lot higher when it was not a holiday. This is consistent with the results that we see in the box plot for the distribution of consumption power between the two groups. The third way we compared and contrasted was did a trend anaylsis on the mean power consumption based on month and year. This showed that there is a general trend that the mean power consumption was a lot higher when it was not a holiday. There are some similar peaks seen in between the years 2020 and 2022 but holidays see much more troughs throughout the years which are more common. We also printed more summary statistics such as mean and standard deviation for holiday or

not, in which the mean, median and standard deviation are all higher when it is not a holiday.

Objective 3

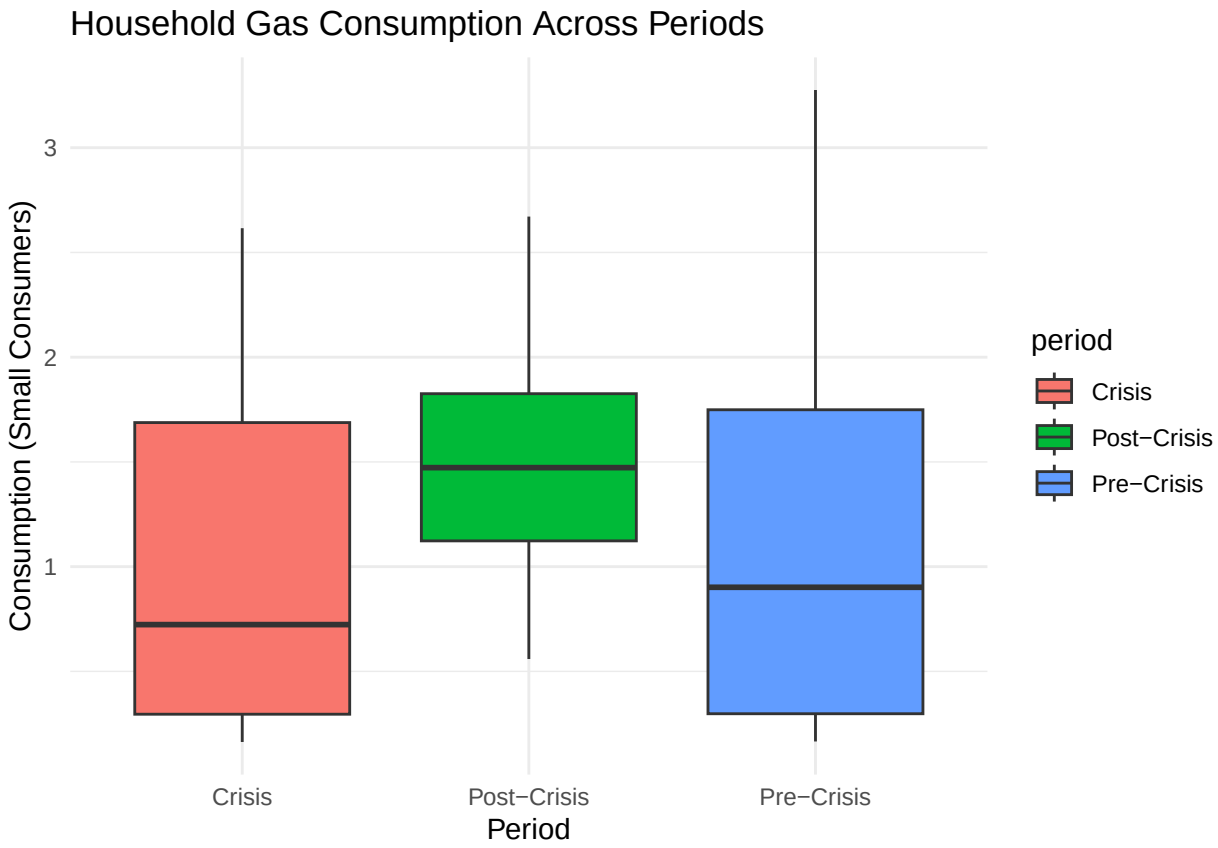
- According to the paper, the gas crisis occurred between September 2021 until October 2022. Compare this period with the periods before and after on household and industrial consumption. Write a paragraph explaining your findings.

```
daily$date <- as.Date(daily$date)

# crisis periods
daily <- daily %>%
  mutate(period = case_when(
    date < as.Date("2021-09-01") ~ "Pre-Crisis",
    date >= as.Date("2021-09-01") & date <= as.Date("2022-10-31") ~ "Crisis",
    date > as.Date("2022-10-31") ~ "Post-Crisis"
  ))

# find mean consumption for each period
consumption_summary <- daily %>%
  group_by(period) %>%
  summarise(
    avg_household = mean(consumption_small, na.rm = TRUE),
    avg_industry = mean(consumption_industry, na.rm = TRUE)
  )

print(consumption_summary)
```

```
## # A tibble: 3 x 3
##   period      avg_household avg_industry
##   <chr>               <dbl>        <dbl>
## 1 Crisis              0.977         1.10
## 2 Post-Crisis         1.53          1.07
## 3 Pre-Crisis          1.07          1.23
```

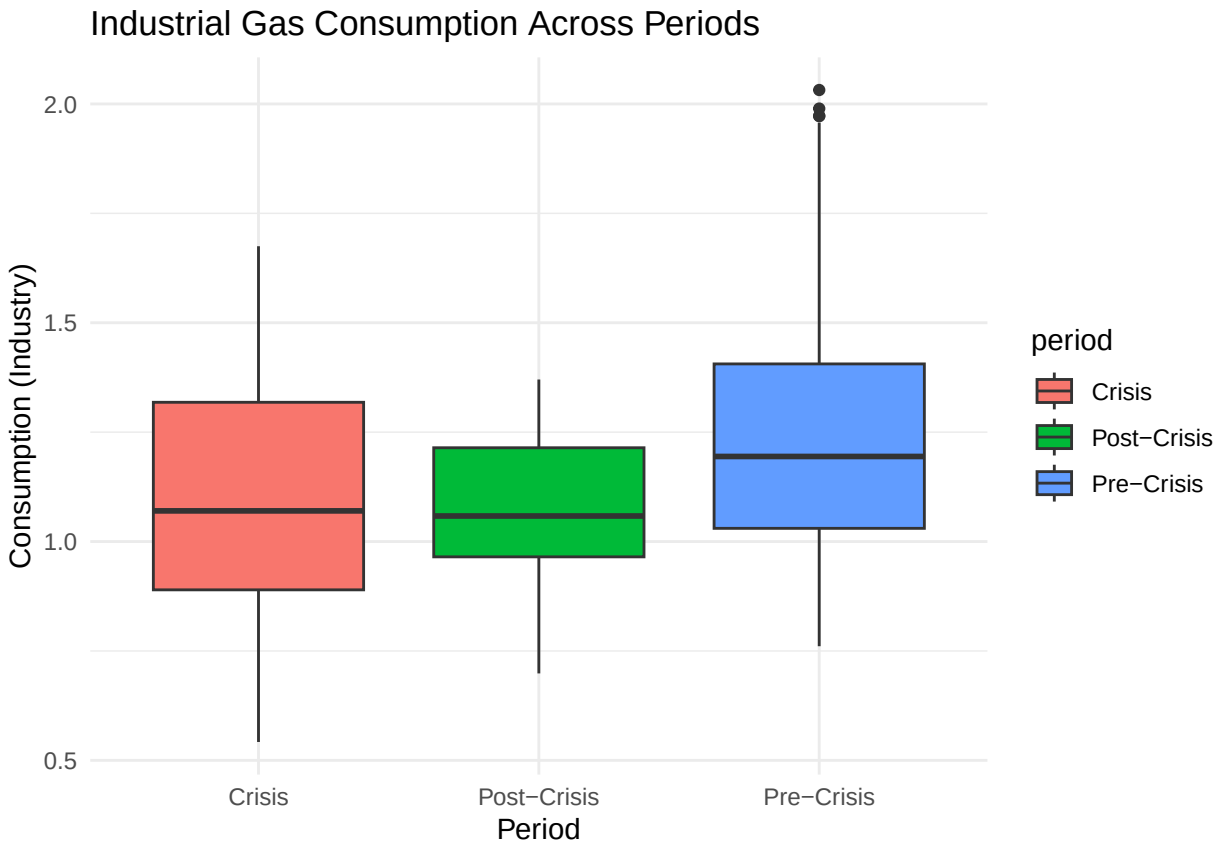
```
ggplot(daily, aes(x = period, y = consumption_small, fill = period)) +
  geom_boxplot() +
  labs(title = "Household Gas Consumption Across Periods",
       x = "Period",
       y = "Consumption (Small Consumers)") +
  theme_minimal()
```

```
## Warning: Removed 365 rows containing non-finite values (`stat_boxplot()`).
```




```
ggplot(daily, aes(x = period, y = consumption_industry, fill = period)) +
  geom_boxplot() +
  labs(title = "Industrial Gas Consumption Across Periods",
       x = "Period",
       y = "Consumption (Industry)") +
  theme_minimal()
```

```
## Warning: Removed 1 rows containing non-finite values (`stat_boxplot()`).
```



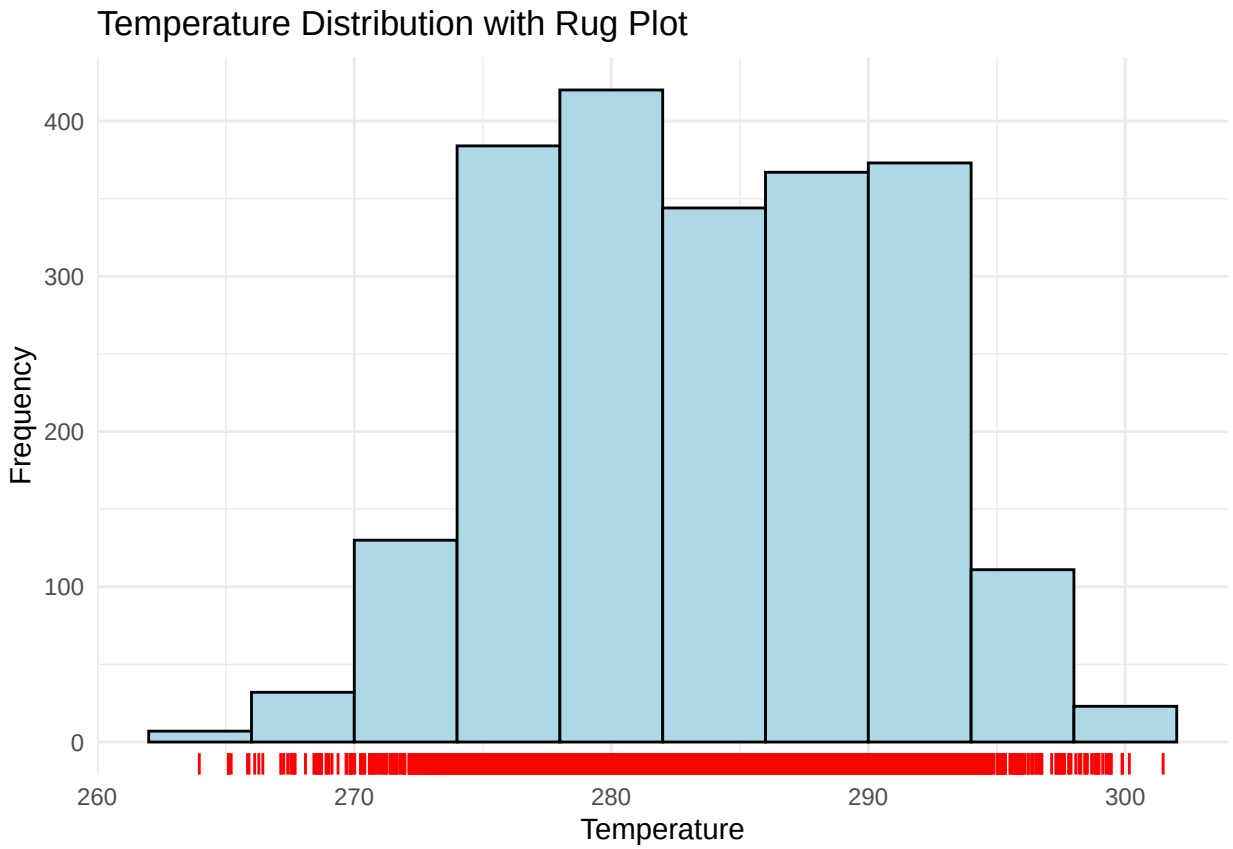

```
#ask about warning
```

Comparing the consumption between households (consumption_small) and industries (consumption_industry) across different periods i.e. pre and post crisis, we see that the average industry consumption was higher pre-crisis and during the crisis when compared to the average consumption by the households. The average household consumption was highest post crisis when compared to itself during the different periods. The average industry consumption was highest pre-crisis when compared to itself. In general, we see that the highest consumption across the time periods and categories is the household consumption post-crisis. The median consumption of households pre-crisis and during crisis is similar with median consumption post-crisis being the highest. The range is, however, the largest pre-crisis. For industry consumptions, the median across periods (pre-crisis, during-crisis, post-crisis) is almost similar, with precrisis, being slightly higher. In general, we see that households have a larger consumption post-crisis whereas mean industry consumption was a lot higher pre and during crisis.

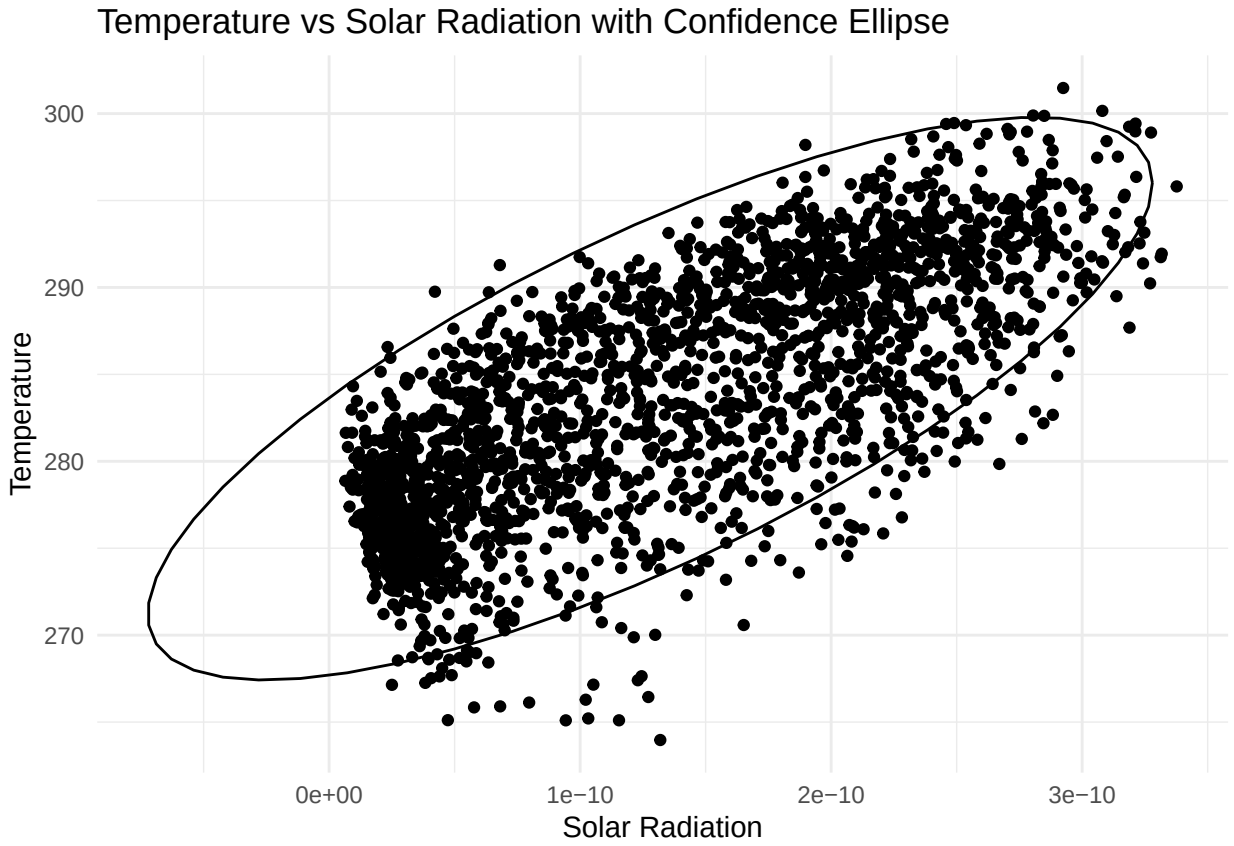
Objective 4

- Explore the documentation for ggplot. Select one geometery and one `stat_` function we have not used before or use an option to a previously used geometry/stat with a new option. Write a short paragraph explaining what the plots show.

```
ggplot(daily, aes(x = temperature)) +
  geom_histogram(binwidth = 4, fill = "lightblue", color = "black") +
  geom_rug(sides = "b", color = "red") +
  labs(title = "Temperature Distribution with Rug Plot", x = "Temperature", y = "Frequency") + theme_mi
```




```
ggplot(daily, aes(x = solar_radiation, y = temperature)) +
  geom_point() +
  stat_ellipse(level = 0.95) +
  labs(title = "Temperature vs Solar Radiation with Confidence Ellipse", x = "Solar Radiation", y = "Te
  theme_minimal()
```



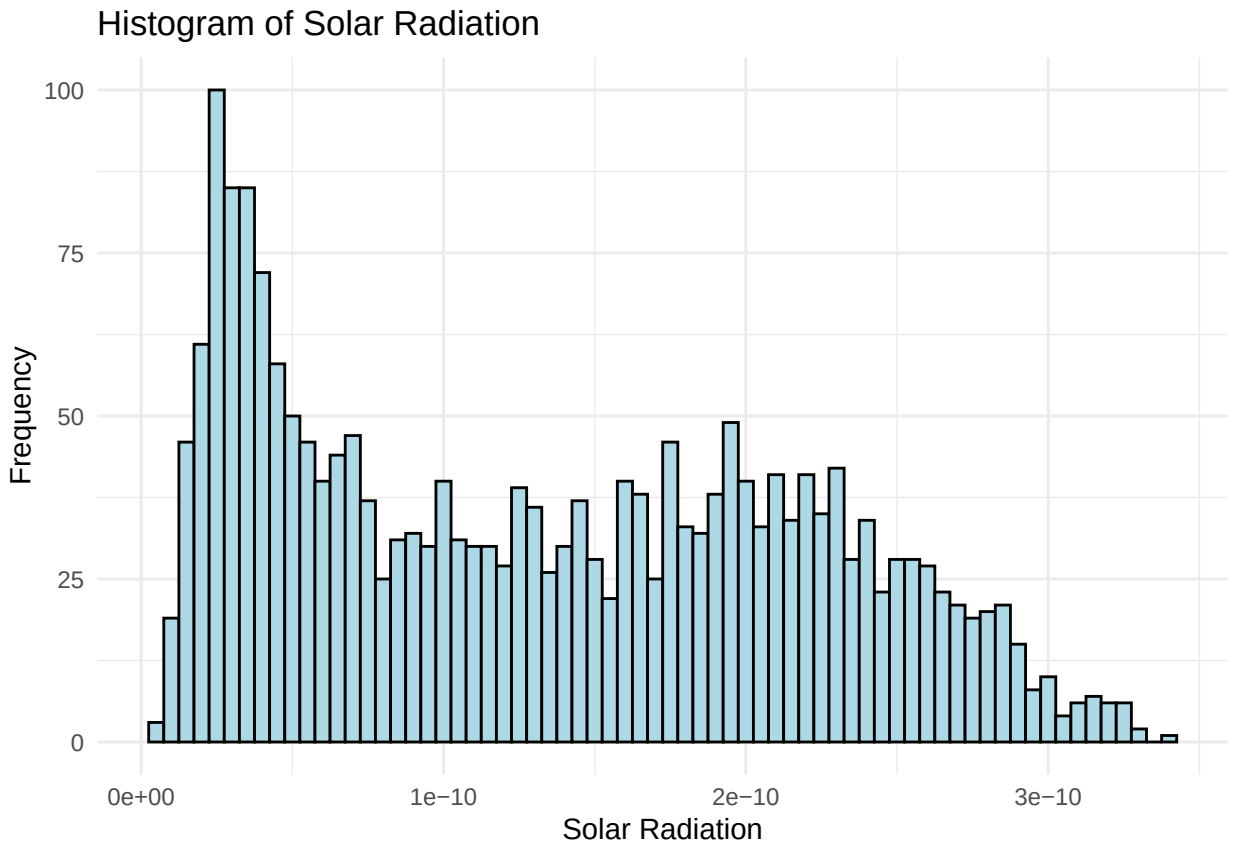
The geom_rug() function adds red ticks on the bottom of the x-axis to represent individual temperature data points. Although the geom_rug() function can be used on its own, adding a histogram aids in visualizing the distribution. We can see that a majority of the temperature values range from about 273 to 295, with fewer values occurring outside this range. These less frequent values are represented by the sparser tick marks at the lower and higher ends of the x-axis.

The stat_ellipse() function shows the distribution of data points in a scatter plot and adds a confidence ellipse. The ellipse represents the region where we expect 95% of the points to lie. In this plot, it visualizes the relationship between solar radiation and temperature, with the ellipse indicating the spread of the data points. The size and shape of the ellipse reveal how the two variables are related: a larger ellipse suggests more variation, while a more circular shape indicates less correlation between the variables. Based on the ellipse in this plot, it aligns well with most of the values, but there is a significant gap at the bottom, indicating that fewer data points fall in that area. We can also see that since the ellipse shape is pretty elongated, there is a positive correlation between temperature and solar radiation.

Objective 4

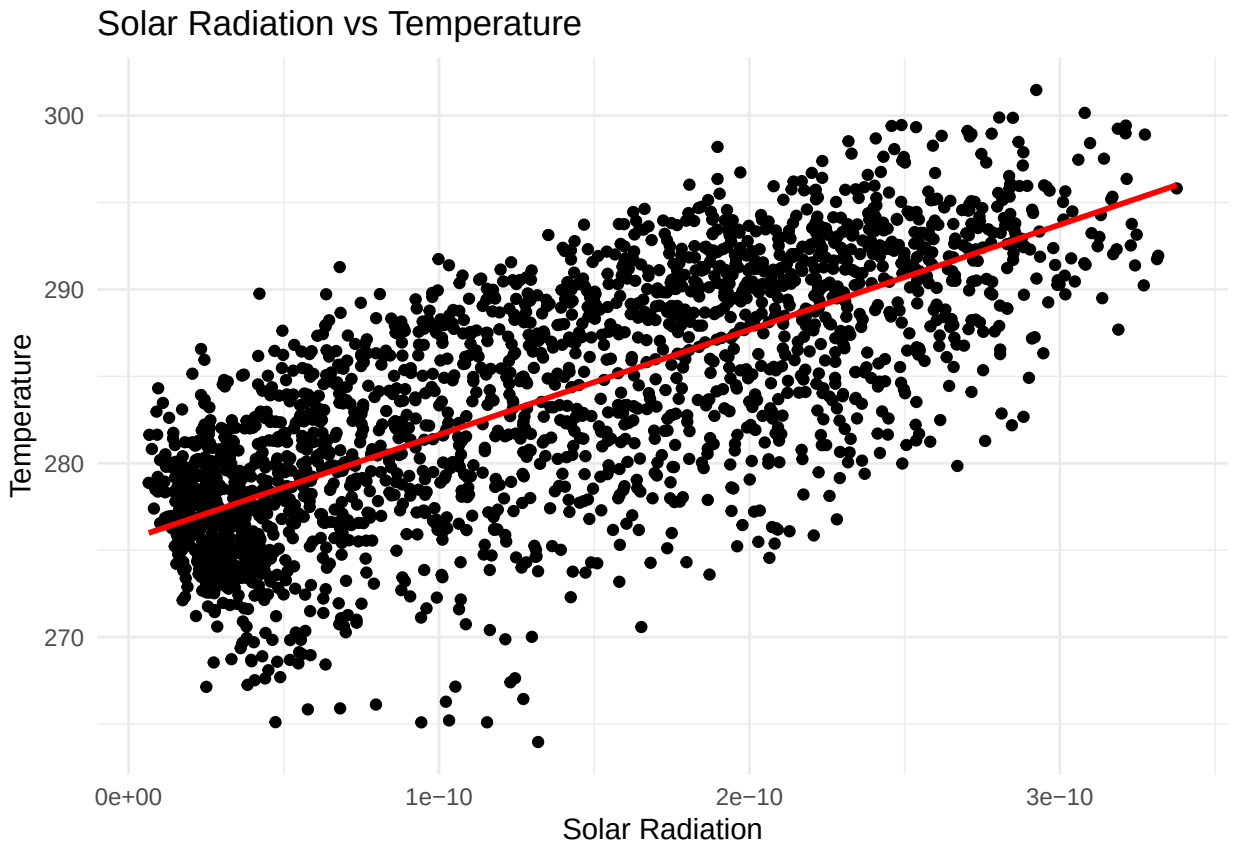
- Investigate solar radiation's marginal distribution and also it's relationship with temperature.

```
ggplot(daily, aes(x = solar_radiation)) +
  geom_histogram( fill = "lightblue", color = "black", binwidth = 5e-12) +
  labs(title = "Histogram of Solar Radiation", x = "Solar Radiation", y = "Frequency") +
  theme_minimal()
```




```
ggplot(daily, aes(x = solar_radiation, y = temperature)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  labs(title = "Solar Radiation vs Temperature", x = "Solar Radiation", y = "Temperature") +
  theme_minimal()
```

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



The marginal distribution of solar radiation is right-skewed, with the highest frequency of values concentrated between 0 and 1e-10. The values remain consistently frequent from 1e-10 to about 2.5e-10 before tapering off.

The relationship between solar radiation and temperature is positively correlated, meaning that as solar radiation increases, temperature also tends to rise. This is expected since more solar radiation typically leads to higher temperatures.

Objective 5

- Use `group_by` to summarize by a new feature of this data set not otherwise discussed in the tasks or objectives. What have you learned with these investigation?

```
daily <- daily |>
  mutate(weekday_name = weekdays(as.Date(date)))

# group by weekday to summarize
weekday_summary <- daily |>
  group_by(weekday_name) |>
  summarize(
    avg_consumption_small = mean(consumption_small, na.rm = TRUE),
    avg_consumption_industry = mean(consumption_industry, na.rm = TRUE),
    avg_consumption_power = mean(consumption_power, na.rm = TRUE)
  ) |>
  arrange(match(weekday_name, c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Su

print(weekday_summary)
```

```
## # A tibble: 7 x 4
##   weekday_name avg_consumption_small avg_consumption_industry
```

```
##   <chr>                          <dbl>                    <dbl>
## 1 Monday                          1.08                     1.25
## 2 Tuesday                         1.07                     1.25
## 3 Wednesday                       1.07                     1.25
## 4 Thursday                        1.07                     1.24
## 5 Friday                          1.06                     1.20
## 6 Saturday                        1.05                     1.08
## 7 Sunday                          1.05                     1.11
## # i 1 more variable: avg_consumption_power <dbl>
```

For both industrial consumption and power consumption, the average consumption is significantly higher on weekdays in comparison to weekends. This may be because industries have reduced hours or or are completely closed on weekdays, decreasing hte average. However, when looking at small consumer consumption, it is relatively consistent (although generally decreasing slightly Monday to Sunday). This is probably because residential needs like heating don't change much depending on day. The very slightly lower values on weekends may be because they're using less gas.

Objective 6

- Based on your exploration of the data, suggest three questions that could be asked from these data or additional data that you can imagine. Be sure to explain why the previous plots or calculations indicates that this would be an interesting our useful exploration.

Objective 7

- Write an abstract for your project. Briefly explain what you did, what you found, and why a potential reader should be interested in your research.