# Watt –to- Weather: Wrangling the Energy and Climate Connection

Authors: Ajinkya Phanse , Darshit Shah

16:954:597:01: Data Wrangling and Husbandry

Prof. Stevenson Bolivar-Atuesta

# TABLE OF CONTENTS

Here’s what you’ll find :

1. Project Overview
2. Importance of the Topic
3. Dataset Overview :
    1. Data Extraction Pipeline
    2. Data Cleaning Process
4. Integration Strategy
5. Exploratory Data Analysis (EDA)
6. Key Findings from EDA
7. ML Component
8. Challenges & Solutions
9. Final Insights
10. Future Work
11. References
12. Thank You + Q&A

# Project Overview

Our data warehouse project analyzes the relationship between weather conditions and energy demand across 20 major U.S. cities, revealing critical patterns that impact energy infrastructure planning and management.

The Challenge

Energy providers face significant challenges forecasting demand fluctuations driven by weather variables, leading to:

- Suboptimal resource allocation during peak demand periods

- Inefficient infrastructure planning

- Missed opportunities for targeted efficiency programs

# Importance of the Topic

*Critical Resource Planning*: Energy demand fluctuates 20-30% based on temperature variations, requiring precise forecasting

*Data Integration Challenge*: Weather and energy data exist in incompatible formats with different:

- Geographic boundaries (city-based vs. regional)
- Time resolutions (hourly vs. daily)
- Measurement standards and units

# Importance of the Topic

***R-Based Data Wrangling Solution***: Our project demonstrates how R's specialized packages overcome these challenges through:

- Cross-domain data integration using tidyverse tools
- Geographic mapping with custom join operations
- Time series normalization with lubridate
- Automated data quality improvement (reduced missing values by 95.7%)

***Business Impact***: Our wrangled dataset enables:

- Precise identification of optimal temperatures (51.5°F)
- Regional sensitivity mapping for targeted infrastructure planning
- Quantifiable metrics for climate adaptation planning
- Significant improvement in demand forecasting accuracy

# Dataset Overview

What we saw when we got here

# Data Sources

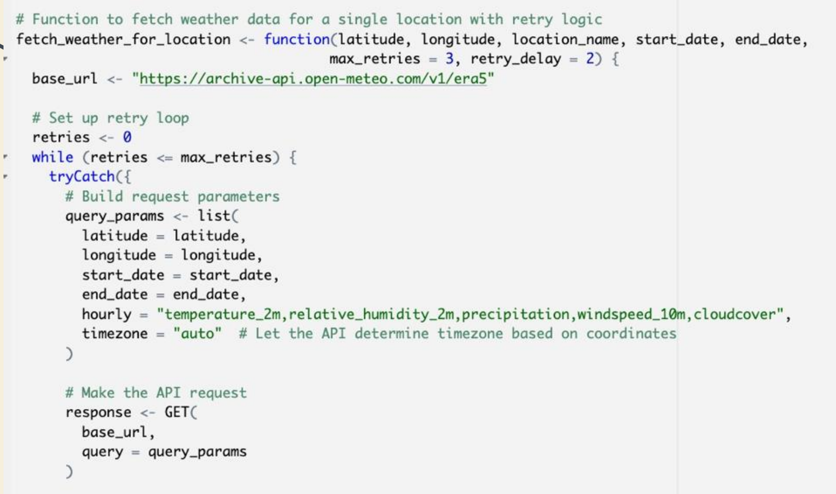
***Weather Data:***

- Source: Open-Meteo ERA5 Historical Weather API (https://archive-api.open-meteo.com/v1/era5)
- Coverage: Hourly data for 20 major U.S. cities throughout 2024
- Variables: Temperature, humidity, precipitation, wind speed, cloud cover
- Extraction: Custom R functions with retry logic and rate limit handling
- Volume: ·180k records processed
- Key Challenge: Standardizing location formats for regional mapping

***Energy Data:***

- Source: U.S. Energy Information Administration (EIA) API v2 (https://api.eia.gov/v2/electricity/rto/daily-region-data/data/)
- Coverage: Daily regional energy data from major U.S. interconnections
- Variables: Demand, generation, interchange values across utility companies
- Extraction: Paginated API requests with 500 records/day sampling
- Volume: ·180k daily energy records across multiple regions
- Key Challenge: Mapping energy regions to weather observation locations

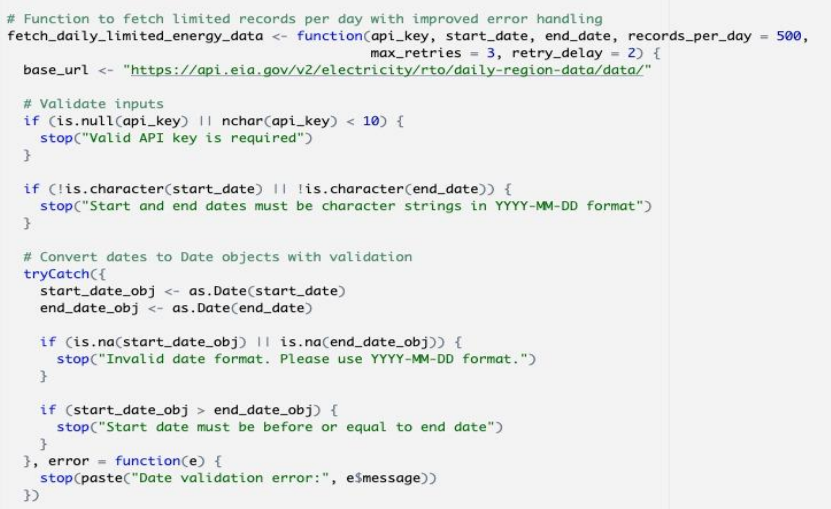
# Data Extraction Pipeline



```r
# Function to fetch weather data for a single location with retry logic
fetch_weather_for_location <- function(latitude, longitude, location_name, start_date, end_date,
                                       max_retries = 3, retry_delay = 2) {
  base_url <- "https://archive-api.open-meteo.com/v1/era5"

  # Set up retry loop
  retries <- 0
  while (retries <= max_retries) {
    tryCatch({
      # Build request parameters
      query_params <- list(
        latitude = latitude,
        longitude = longitude,
        start_date = start_date,
        end_date = end_date,
        hourly = "temperature_2m,relative_humidity_2m,precipitation,windspeed_10m,cloudcover",
        timezone = "auto"  # Let the API determine timezone based on coordinates
      )

      # Make the API request
      response <- GET(
        base_url,
        query = query_params
      )
```

*Weather* : Uses httr for API requests, implements tryCatch for error handling, and converts JSON responses to tidy tibbles with standardized column names for consistent downstream analysis.

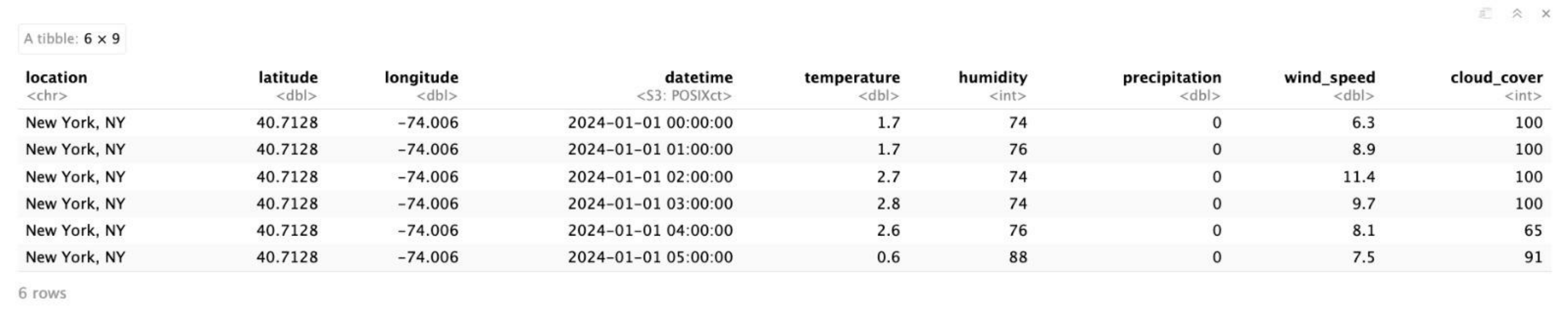
*Energy* : Uses for-loops with pagination to systematically extract data, leverages bind_rows to combine multiple API responses, and employs date manipulation functions from lubridate for temporal data management.



```r
# Function to fetch limited records per day with improved error handling
fetch_daily_limited_energy_data <- function(api_key, start_date, end_date, records_per_day = 500,
                                            max_retries = 3, retry_delay = 2) {
  base_url <- "https://api.eia.gov/v2/electricity/rto/daily-region-data/data/"

  # Validate inputs
  if (is.null(api_key) || nchar(api_key) < 10) {
    stop("Valid API key is required")
  }

  if (!is.character(start_date) || !is.character(end_date)) {
    stop("Start and end dates must be character strings in YYYY-MM-DD format")
  }

  # Convert dates to Date objects with validation
  tryCatch({
    start_date_obj <- as.Date(start_date)
    end_date_obj <- as.Date(end_date)

    if (is.na(start_date_obj) || is.na(end_date_obj)) {
      stop("Invalid date format. Please use YYYY-MM-DD format.")
    }

    if (start_date_obj > end_date_obj) {
      stop("Start date must be before or equal to end date")
    }
  }, error = function(e) {
    stop(paste("Date validation error:", e$message))
  })
```

# Tables of Data

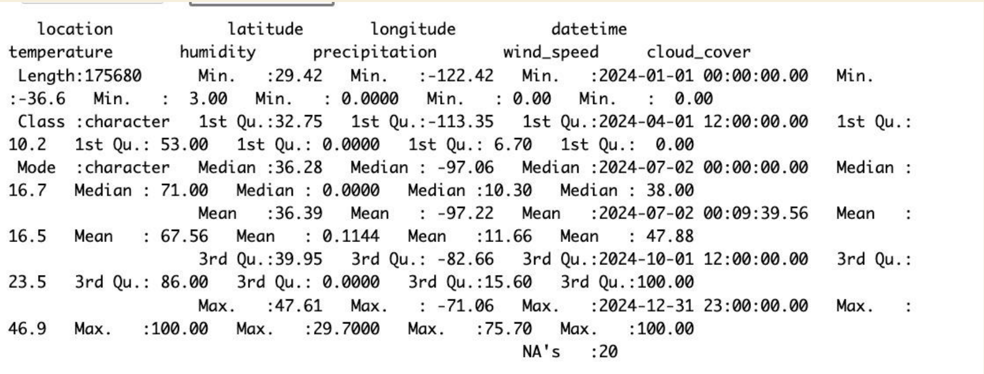
1) Raw Weather Data:



A tibble: 6 x 9

| location<br><chr> | latitude<br><dbl> | longitude<br><dbl> | datetime<br><S3: POSIXct> | temperature<br><dbl> | humidity<br><int> | precipitation<br><dbl> | wind_speed<br><dbl> | cloud_cover<br><int> |
|---|---|---|---|---|---|---|---|---|
| New York, NY | 40.7128 | -74.006 | 2024-01-01 00:00:00 | 1.7 | 74 | 0 | 6.3 | 100 |
| New York, NY | 40.7128 | -74.006 | 2024-01-01 01:00:00 | 1.7 | 76 | 0 | 8.9 | 100 |
| New York, NY | 40.7128 | -74.006 | 2024-01-01 02:00:00 | 2.7 | 74 | 0 | 11.4 | 100 |
| New York, NY | 40.7128 | -74.006 | 2024-01-01 03:00:00 | 2.8 | 74 | 0 | 9.7 | 100 |
| New York, NY | 40.7128 | -74.006 | 2024-01-01 04:00:00 | 2.6 | 76 | 0 | 8.1 | 65 |
| New York, NY | 40.7128 | -74.006 | 2024-01-01 05:00:00 | 0.6 | 88 | 0 | 7.5 | 91 |

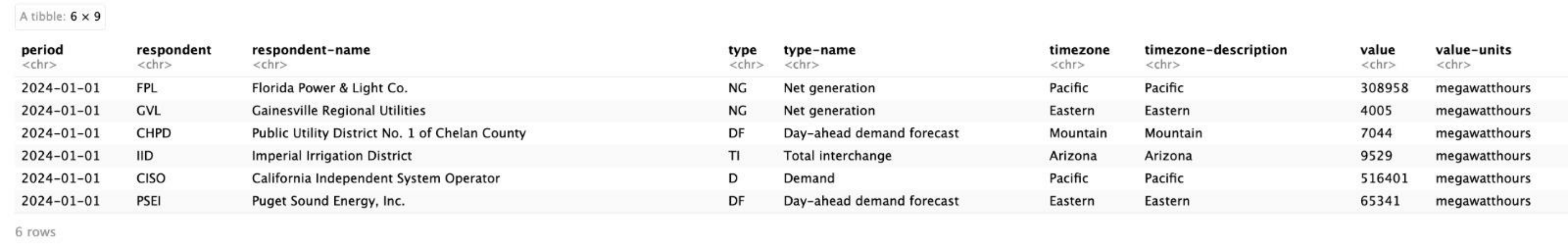
6 rows



```
  location            latitude        longitude          datetime
temperature        humidity      precipitation       wind_speed      cloud_cover
 Length:175680      Min.   :29.42   Min.   :-122.42   Min.   :2024-01-01 00:00:00.00   Min.
:-36.6   Min.   :  3.00   Min.   : 0.0000   Min.   : 0.00   Min.   :  0.00
 Class :character   1st Qu.:32.75   1st Qu.:-113.35   1st Qu.:2024-04-01 12:00:00.00   1st Qu.:
10.2   1st Qu.: 53.00   1st Qu.: 0.0000   1st Qu.: 6.70   1st Qu.:  0.00
 Mode  :character   Median :36.28   Median : -97.06   Median :2024-07-02 00:00:00.00   Median :
16.7   Median : 71.00   Median : 0.0000   Median :10.30   Median : 38.00
                    Mean   :36.39   Mean   : -97.22   Mean   :2024-07-02 00:09:39.56   Mean   :
16.5   Mean   : 67.56   Mean   : 0.1144   Mean   :11.66   Mean   : 47.88
                    3rd Qu.:39.95   3rd Qu.: -82.66   3rd Qu.:2024-10-01 12:00:00.00   3rd Qu.:
23.5   3rd Qu.: 86.00   3rd Qu.: 0.0000   3rd Qu.:15.60   3rd Qu.:100.00
                    Max.   :47.61   Max.   : -71.06   Max.   :2024-12-31 23:00:00.00   Max.   :
46.9   Max.   :100.00   Max.   :29.7000   Max.   :75.70   Max.   :100.00
                                                            NA's   :20
```

# Tables of Data

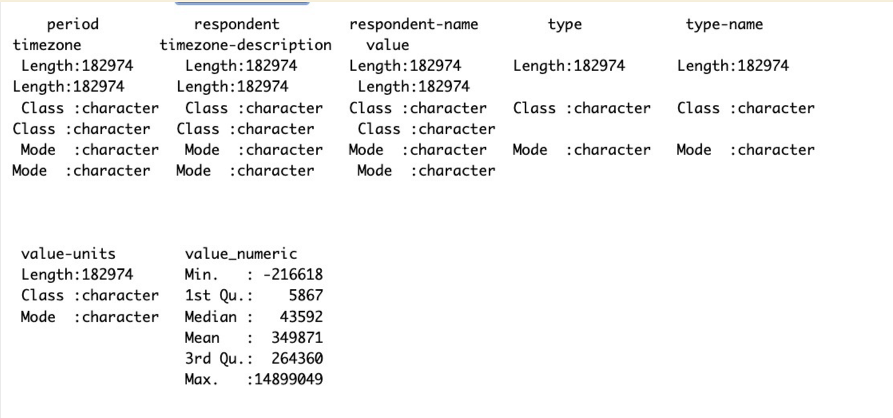
1) Raw Energy Data:



A tibble: 6 × 9

| period<br>&lt;chr&gt; | respondent<br>&lt;chr&gt; | respondent-name<br>&lt;chr&gt; | type<br>&lt;chr&gt; | type-name<br>&lt;chr&gt; | timezone<br>&lt;chr&gt; | timezone-description<br>&lt;chr&gt; | value<br>&lt;chr&gt; | value-units<br>&lt;chr&gt; |
|---|---|---|---|---|---|---|---|---|
| 2024-01-01 | FPL | Florida Power & Light Co. | NG | Net generation | Pacific | Pacific | 308958 | megawatthours |
| 2024-01-01 | GVL | Gainesville Regional Utilities | NG | Net generation | Eastern | Eastern | 4005 | megawatthours |
| 2024-01-01 | CHPD | Public Utility District No. 1 of Chelan County | DF | Day-ahead demand forecast | Mountain | Mountain | 7044 | megawatthours |
| 2024-01-01 | IID | Imperial Irrigation District | TI | Total interchange | Arizona | Arizona | 9529 | megawatthours |
| 2024-01-01 | CISO | California Independent System Operator | D | Demand | Pacific | Pacific | 516401 | megawatthours |
| 2024-01-01 | PSEI | Puget Sound Energy, Inc. | DF | Day-ahead demand forecast | Eastern | Eastern | 65341 | megawatthours |

6 rows



```
    period             respondent        respondent-name        type            type-name       
timezone          timezone-description    value
 Length:182974      Length:182974      Length:182974      Length:182974      Length:182974     
Length:182974      Length:182974       Length:182974     
 Class :character   Class :character   Class :character   Class :character   Class :character  
Class :character   Class :character    Class :character  
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character  
Mode  :character   Mode  :character    Mode  :character  



 value-units        value_numeric    
 Length:182974      Min.   : -216618  
 Class :character   1st Qu.:    5867  
 Mode  :character   Median :   43592  
                    Mean   :  349871  
                    3rd Qu.:  264360  
                    Max.   :14899049  
```

# Data Cleaning Process (weather)

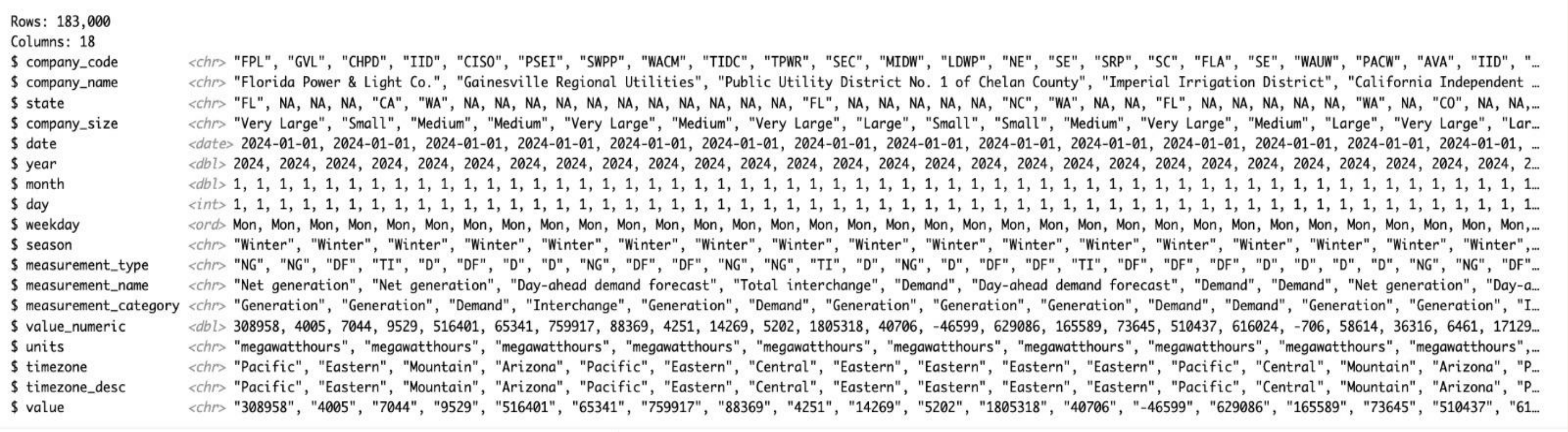
- *Duplicate removal*: Applied distinct() to eliminate identical weather observations that occurred due to API response duplications
- *Column standardization*: Renamed all columns using a consistent naming convention (e.g., temperature_c, humidity_pct) for enhanced code readability and maintainability
- *Temporal feature extraction*: Used lubridate functions to extract date components and create calendar-based features like season, day of week, and time of day variables
- *Feature engineering*: Created derived weather variables such as heat index, temperature range, and extreme weather flags to capture thermal comfort aspects
- *Data categorization*: Developed categorical variables for precipitation intensity, wind speed (using Beaufort scale), and overall weather conditions for easier analysis
- *Geographic enrichment*: Extracted state information from location names and added city-level identifiers for regional analysis
- *NA handling*: Implemented consistent missing value strategies across all data cleaning operations with appropriate na.rm parameters
- *Outlier identification*: Applied statistical methods (z-scores) to flag extreme values for each weather variable
- *Daily aggregation*: Created daily summary statistics from hourly observations using group_by() and summarise() functions, calculating metrics like daily min/max temperatures and total precipitation

# Data Cleaning Process (energy)

- *Duplicate elimination*: Removed exact duplicate energy records that occurred from repeated API calls and pagination overlap
- *Variable standardization*: Transformed character-type values to appropriate numeric formats for statistical analysis using as.numeric() conversion
- *Geographical mapping*: Created a custom mapping system to associate company codes with states and regions based on utility service territories
- *Company categorization*: Classified energy companies by size and type using domain knowledge and average production/demand values
- *Temporal feature creation*: Added date-based variables including season, quarter, and day type (weekday/weekend) to enable time-based analysis
- *Measurement categorization*: Developed a taxonomy for energy measurements by grouping similar types (generation, demand, interchange) for consistent analysis
- *Regional aggregation*: Created region-level summaries from company-level data to align with weather observation regions
- *Outlier detection*: Identified extreme values using statistical thresholds and flagged them with logical indicators
- *Derived variables*: Calculated percentage changes, ratios between generation and demand, and other metrics to support analytical insights
- *Missing value handling*: Applied consistent strategies for NA values using complete cases and appropriate imputation where necessary

# Shocking Result

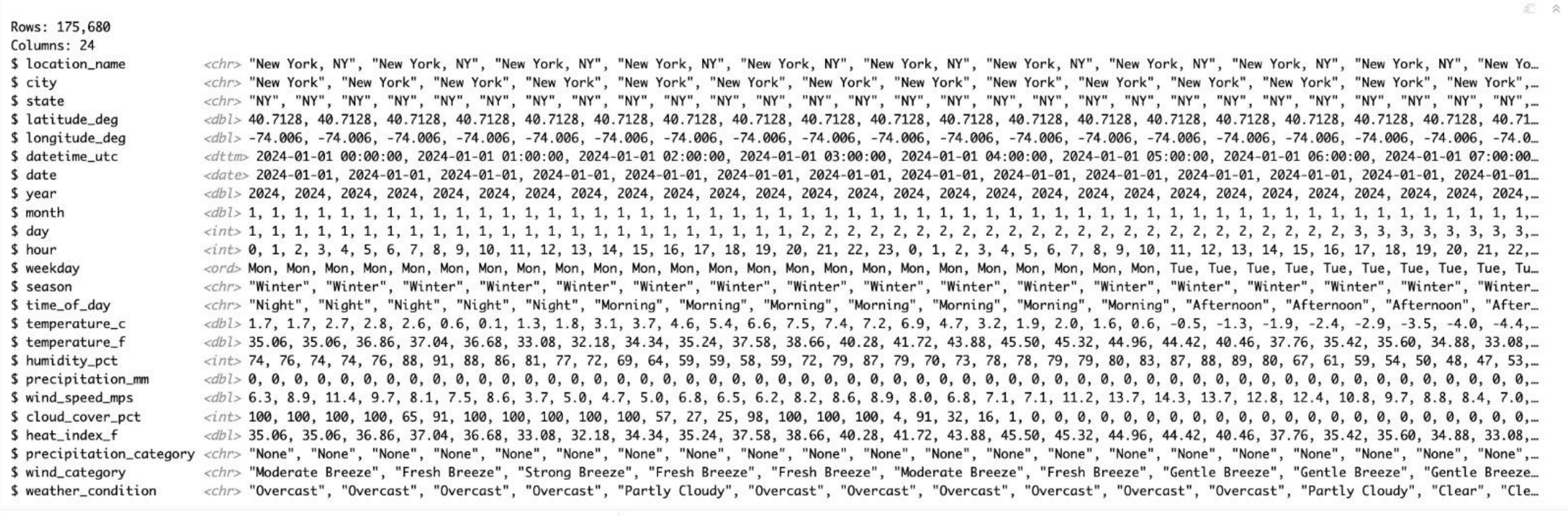
Energy Data (after cleaning)



```
Rows: 183,000
Columns: 18
$ company_code         <chr> "FPL", "GVL", "CHPD", "IID", "CISO", "PSEI", "SWPP", "WACM", "TIDC", "TPWR", "SEC", "MIDW", "LDWP", "NE", "SE", "SRP", "SC", "FLA", "SE", "WAUW", "PACW", "AVA", "IID", "…
$ company_name         <chr> "Florida Power & Light Co.", "Gainesville Regional Utilities", "Public Utility District No. 1 of Chelan County", "Imperial Irrigation District", "California Independent …
$ state                <chr> "FL", NA, NA, NA, "CA", "WA", NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, "FL", NA, NA, NA, NA, NA, "NC", "WA", NA, NA, "FL", NA, NA, NA, NA, NA, "WA", NA, "CO", NA, NA,…
$ company_size         <chr> "Very Large", "Small", "Medium", "Medium", "Very Large", "Medium", "Very Large", "Large", "Small", "Small", "Medium", "Very Large", "Medium", "Large", "Very Large", "Lar…
$ date                 <date> 2024-01-01, 2024-01-01, 2024-01-01, 2024-01-01, 2024-01-01, 2024-01-01, 2024-01-01, 2024-01-01, 2024-01-01, 2024-01-01, 2024-01-01, 2024-01-01, 2024-01-01, 2024-01-01, …
$ year                 <dbl> 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2…
$ month                <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…
$ day                  <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…
$ weekday              <ord> Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon,…
$ season               <chr> "Winter", "Winter", "Winter", "Winter", "Winter", "Winter", "Winter", "Winter", "Winter", "Winter", "Winter", "Winter", "Winter", "Winter", "Winter", "Winter", "Winter",…
$ measurement_type     <chr> "NG", "NG", "DF", "TI", "D", "DF", "D", "D", "NG", "DF", "DF", "NG", "NG", "TI", "D", "NG", "D", "DF", "DF", "TI", "DF", "DF", "DF", "D", "D", "D", "D", "NG", "NG", "DF"…
$ measurement_name     <chr> "Net generation", "Net generation", "Day-ahead demand forecast", "Total interchange", "Demand", "Day-ahead demand forecast", "Demand", "Demand", "Net generation", "Day-a…
$ measurement_category <chr> "Generation", "Generation", "Demand", "Interchange", "Generation", "Demand", "Generation", "Generation", "Generation", "Demand", "Demand", "Generation", "Generation", "I…
$ value_numeric        <dbl> 308958, 4005, 7044, 9529, 516401, 65341, 759917, 88369, 4251, 14269, 5202, 1805318, 40706, -46599, 629086, 165589, 73645, 510437, 616024, -706, 58614, 36316, 6461, 17129…
$ units                <chr> "megawatthours", "megawatthours", "megawatthours", "megawatthours", "megawatthours", "megawatthours", "megawatthours", "megawatthours", "megawatthours", "megawatthours",…
$ timezone             <chr> "Pacific", "Eastern", "Mountain", "Arizona", "Pacific", "Eastern", "Central", "Eastern", "Eastern", "Eastern", "Eastern", "Pacific", "Central", "Mountain", "Arizona", "P…
$ timezone_desc        <chr> "Pacific", "Eastern", "Mountain", "Arizona", "Pacific", "Eastern", "Central", "Eastern", "Eastern", "Eastern", "Eastern", "Pacific", "Central", "Mountain", "Arizona", "P…
$ value                <chr> "308958", "4005", "7044", "9529", "516401", "65341", "759917", "88369", "4251", "14269", "5202", "1805318", "40706", "-46599", "629086", "165589", "73645", "510437", "61…
```

# Shocking Result

Weather Data (after cleaning)



```
Rows: 175,680
Columns: 24
$ location_name          <chr> "New York, NY", "New York, NY", "New York, NY", "New York, NY", "New York, NY", "New York, NY", "New York, NY", "New York, NY", "New York, NY", "New York, NY", "New Yo…
$ city                   <chr> "New York", "New York", "New York", "New York", "New York", "New York", "New York", "New York", "New York", "New York", "New York", "New York", "New York", "New York",…
$ state                  <chr> "NY", "NY", "NY", "NY", "NY", "NY", "NY", "NY", "NY", "NY", "NY", "NY", "NY", "NY", "NY", "NY", "NY", "NY", "NY", "NY", "NY", "NY", "NY", "NY", "NY", "NY", "NY", "NY",…
$ latitude_deg           <dbl> 40.7128, 40.7128, 40.7128, 40.7128, 40.7128, 40.7128, 40.7128, 40.7128, 40.7128, 40.7128, 40.7128, 40.7128, 40.7128, 40.7128, 40.7128, 40.7128, 40.7128, 40.7128, 40.71…
$ longitude_deg          <dbl> -74.006, -74.006, -74.006, -74.006, -74.006, -74.006, -74.006, -74.006, -74.006, -74.006, -74.006, -74.006, -74.006, -74.006, -74.006, -74.006, -74.006, -74.006, -74.0…
$ datetime_utc           <dttm> 2024-01-01 00:00:00, 2024-01-01 01:00:00, 2024-01-01 02:00:00, 2024-01-01 03:00:00, 2024-01-01 04:00:00, 2024-01-01 05:00:00, 2024-01-01 06:00:00, 2024-01-01 07:00:00…
$ date                   <date> 2024-01-01, 2024-01-01, 2024-01-01, 2024-01-01, 2024-01-01, 2024-01-01, 2024-01-01, 2024-01-01, 2024-01-01, 2024-01-01, 2024-01-01, 2024-01-01, 2024-01-01, 2024-01-01…
$ year                   <dbl> 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024, 2024,…
$ month                  <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,…
$ day                    <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3,…
$ hour                   <int> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22,…
$ weekday                <ord> Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Mon, Tue, Tue, Tue, Tue, Tue, Tue, Tue, Tue, Tue, Tu…
$ season                 <chr> "Winter", "Winter", "Winter", "Winter", "Winter", "Winter", "Winter", "Winter", "Winter", "Winter", "Winter", "Winter", "Winter", "Winter", "Winter", "Winter", "Winter…
$ time_of_day            <chr> "Night", "Night", "Night", "Night", "Night", "Morning", "Morning", "Morning", "Morning", "Morning", "Morning", "Morning", "Afternoon", "Afternoon", "Afternoon", "After…
$ temperature_c          <dbl> 1.7, 1.7, 2.7, 2.8, 2.6, 0.6, 0.1, 1.3, 1.8, 3.1, 3.7, 4.6, 5.4, 6.6, 7.5, 7.4, 7.2, 6.9, 4.7, 3.2, 1.9, 2.0, 1.6, 0.6, -0.5, -1.3, -1.9, -2.4, -2.9, -3.5, -4.0, -4.4,…
$ temperature_f          <dbl> 35.06, 35.06, 36.86, 37.04, 36.68, 33.08, 32.18, 34.34, 35.24, 37.58, 38.66, 40.28, 41.72, 43.88, 45.50, 45.32, 44.96, 44.42, 40.46, 37.76, 35.42, 35.60, 34.88, 33.08,…
$ humidity_pct           <int> 74, 76, 74, 74, 76, 88, 91, 88, 86, 81, 77, 72, 69, 64, 59, 59, 58, 59, 72, 79, 87, 79, 70, 73, 78, 78, 79, 79, 80, 83, 87, 88, 89, 80, 67, 61, 59, 54, 50, 48, 47, 53,…
$ precipitation_mm       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,…
$ wind_speed_mps         <dbl> 6.3, 8.9, 11.4, 9.7, 8.1, 7.5, 8.6, 3.7, 5.0, 4.7, 5.0, 6.8, 6.5, 6.2, 8.2, 8.6, 8.9, 8.0, 6.8, 7.1, 7.1, 11.2, 13.7, 14.3, 13.7, 12.8, 12.4, 10.8, 9.7, 8.8, 8.4, 7.0,…
$ cloud_cover_pct        <int> 100, 100, 100, 100, 65, 91, 100, 100, 100, 100, 100, 57, 27, 25, 98, 100, 100, 100, 4, 91, 32, 16, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,…
$ heat_index_f           <dbl> 35.06, 35.06, 36.86, 37.04, 36.68, 33.08, 32.18, 34.34, 35.24, 37.58, 38.66, 40.28, 41.72, 43.88, 45.50, 45.32, 44.96, 44.42, 40.46, 37.76, 35.42, 35.60, 34.88, 33.08,…
$ precipitation_category <chr> "None", "None", "None", "None", "None", "None", "None", "None", "None", "None", "None", "None", "None", "None", "None", "None", "None", "None", "None", "None", "None",…
$ wind_category          <chr> "Moderate Breeze", "Fresh Breeze", "Strong Breeze", "Fresh Breeze", "Fresh Breeze", "Moderate Breeze", "Fresh Breeze", "Gentle Breeze", "Gentle Breeze", "Gentle Breeze…
$ weather_condition      <chr> "Overcast", "Overcast", "Overcast", "Overcast", "Partly Cloudy", "Overcast", "Overcast", "Overcast", "Overcast", "Overcast", "Overcast", "Partly Cloudy", "Clear", "Cle…
```

# Integration Strategy

- Created a custom mapping table connecting weather cities to energy regions based on utility service territories
- Executed a two-stage merge process using dplyr's joining functions:

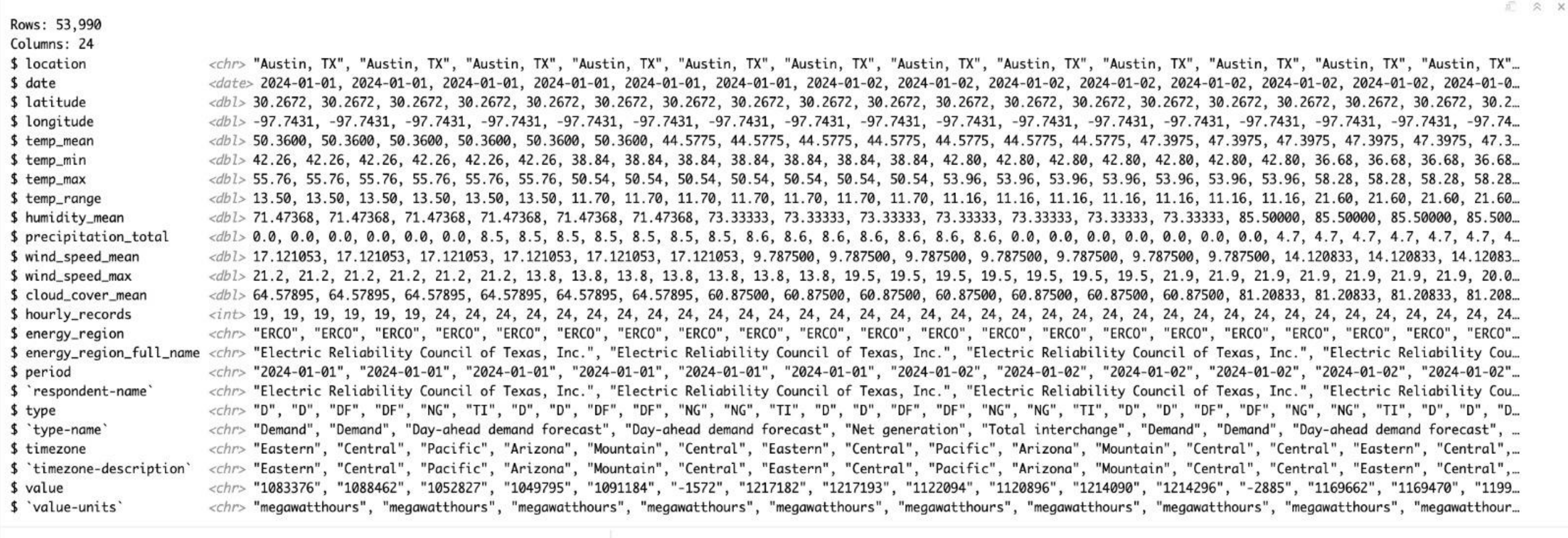
  -Stage 1: Used left_join() to connect daily weather observations with the mapping table
  -Preserved all weather records while adding corresponding energy region codes
  -Join column: "location”

  -Stage 2: Used inner_join() to combine the intermediate dataset with energy data
  -Join columns: "date" and "energy_region"
  -Ensured proper temporal and spatial alignment

- Specified relationship="many-to-many" parameter to handle multiple energy measurements per date-region combination which is critical for preserving demand, generation, and interchange records for each location
- Performed post-merge validation checks:

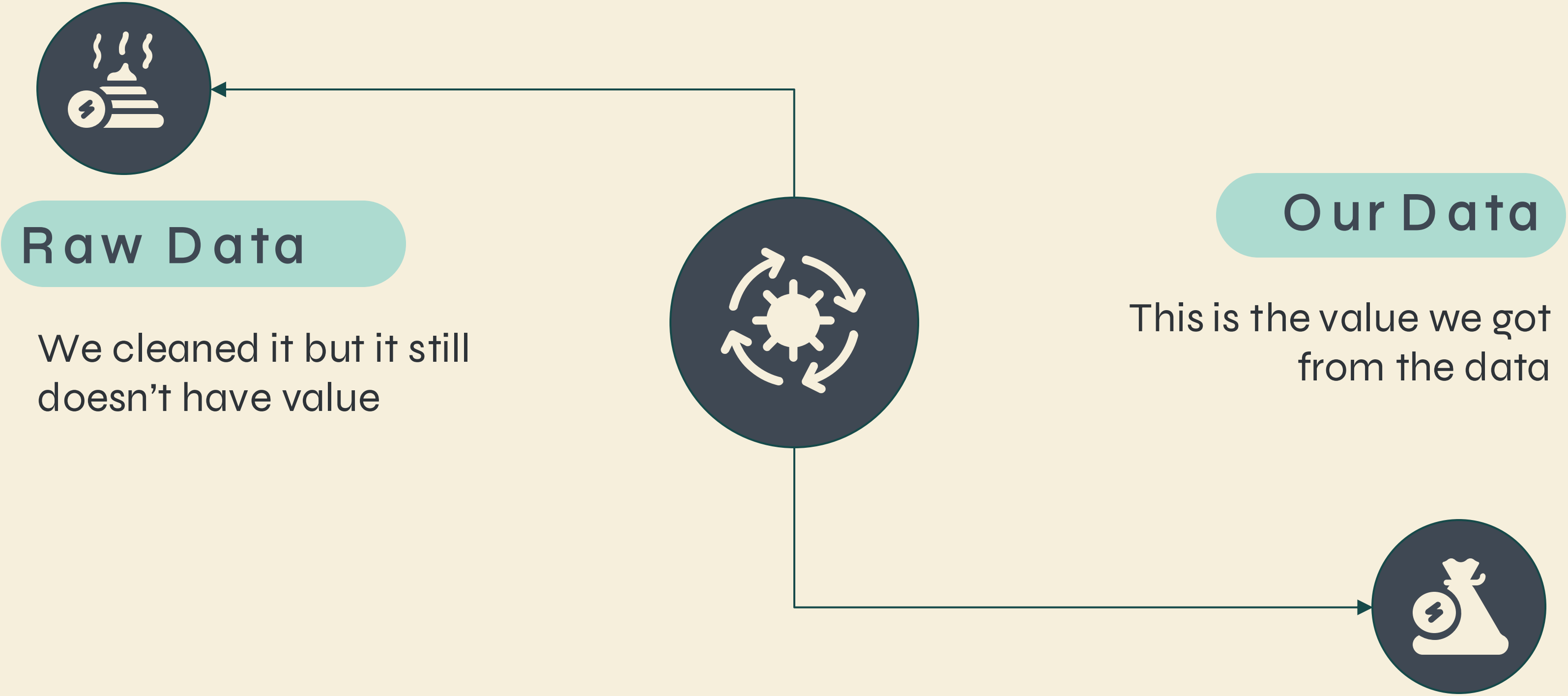
  -Counted unique locations, regions, and dates
  -Verified data coverage against expected values
  -Ensured appropriate data integrity for analysis

# Integration Result



```
Rows: 53,990
Columns: 24
$ location                <chr> "Austin, TX", "Austin, TX", "Austin, TX", "Austin, TX", "Austin, TX", "Austin, TX", "Austin, TX", "Austin, TX", "Austin, TX", "Austin, TX", "Austin, TX", "Austin, TX"…
$ date                    <date> 2024-01-01, 2024-01-01, 2024-01-01, 2024-01-01, 2024-01-01, 2024-01-01, 2024-01-02, 2024-01-02, 2024-01-02, 2024-01-02, 2024-01-02, 2024-01-02, 2024-01-02, 2024-01-0…
$ latitude                <dbl> 30.2672, 30.2672, 30.2672, 30.2672, 30.2672, 30.2672, 30.2672, 30.2672, 30.2672, 30.2672, 30.2672, 30.2672, 30.2672, 30.2672, 30.2672, 30.2672, 30.2672, 30.2672, 30.2…
$ longitude               <dbl> -97.7431, -97.7431, -97.7431, -97.7431, -97.7431, -97.7431, -97.7431, -97.7431, -97.7431, -97.7431, -97.7431, -97.7431, -97.7431, -97.7431, -97.7431, -97.7431, -97.74…
$ temp_mean               <dbl> 50.3600, 50.3600, 50.3600, 50.3600, 50.3600, 50.3600, 44.5775, 44.5775, 44.5775, 44.5775, 44.5775, 44.5775, 44.5775, 47.3975, 47.3975, 47.3975, 47.3975, 47.3975, 47.3…
$ temp_min                <dbl> 42.26, 42.26, 42.26, 42.26, 42.26, 42.26, 38.84, 38.84, 38.84, 38.84, 38.84, 38.84, 38.84, 42.80, 42.80, 42.80, 42.80, 42.80, 42.80, 42.80, 36.68, 36.68, 36.68, 36.68…
$ temp_max                <dbl> 55.76, 55.76, 55.76, 55.76, 55.76, 55.76, 50.54, 50.54, 50.54, 50.54, 50.54, 50.54, 50.54, 53.96, 53.96, 53.96, 53.96, 53.96, 53.96, 53.96, 58.28, 58.28, 58.28, 58.28…
$ temp_range              <dbl> 13.50, 13.50, 13.50, 13.50, 13.50, 13.50, 11.70, 11.70, 11.70, 11.70, 11.70, 11.70, 11.70, 11.16, 11.16, 11.16, 11.16, 11.16, 11.16, 11.16, 21.60, 21.60, 21.60, 21.60…
$ humidity_mean           <dbl> 71.47368, 71.47368, 71.47368, 71.47368, 71.47368, 71.47368, 73.33333, 73.33333, 73.33333, 73.33333, 73.33333, 73.33333, 73.33333, 85.50000, 85.50000, 85.50000, 85.500…
$ precipitation_total     <dbl> 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 8.5, 8.5, 8.5, 8.5, 8.5, 8.5, 8.5, 8.6, 8.6, 8.6, 8.6, 8.6, 8.6, 8.6, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 4.7, 4.7, 4.7, 4.7, 4.7, 4.7, 4…
$ wind_speed_mean         <dbl> 17.121053, 17.121053, 17.121053, 17.121053, 17.121053, 17.121053, 9.787500, 9.787500, 9.787500, 9.787500, 9.787500, 9.787500, 9.787500, 14.120833, 14.120833, 14.12083…
$ wind_speed_max          <dbl> 21.2, 21.2, 21.2, 21.2, 21.2, 21.2, 13.8, 13.8, 13.8, 13.8, 13.8, 13.8, 13.8, 19.5, 19.5, 19.5, 19.5, 19.5, 19.5, 19.5, 21.9, 21.9, 21.9, 21.9, 21.9, 21.9, 21.9, 20.0…
$ cloud_cover_mean        <dbl> 64.57895, 64.57895, 64.57895, 64.57895, 64.57895, 64.57895, 60.87500, 60.87500, 60.87500, 60.87500, 60.87500, 60.87500, 60.87500, 81.20833, 81.20833, 81.20833, 81.208…
$ hourly_records          <int> 19, 19, 19, 19, 19, 19, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24…
$ energy_region           <chr> "ERCO", "ERCO", "ERCO", "ERCO", "ERCO", "ERCO", "ERCO", "ERCO", "ERCO", "ERCO", "ERCO", "ERCO", "ERCO", "ERCO", "ERCO", "ERCO", "ERCO", "ERCO", "ERCO", "ERCO", "ERCO"…
$ energy_region_full_name <chr> "Electric Reliability Council of Texas, Inc.", "Electric Reliability Council of Texas, Inc.", "Electric Reliability Council of Texas, Inc.", "Electric Reliability Cou…
$ period                  <chr> "2024-01-01", "2024-01-01", "2024-01-01", "2024-01-01", "2024-01-01", "2024-01-01", "2024-01-02", "2024-01-02", "2024-01-02", "2024-01-02", "2024-01-02", "2024-01-02"…
$ `respondent-name`       <chr> "Electric Reliability Council of Texas, Inc.", "Electric Reliability Council of Texas, Inc.", "Electric Reliability Council of Texas, Inc.", "Electric Reliability Cou…
$ type                    <chr> "D", "D", "DF", "DF", "NG", "TI", "D", "D", "DF", "DF", "NG", "NG", "TI", "D", "D", "DF", "DF", "NG", "NG", "TI", "D", "D", "DF", "DF", "NG", "NG", "TI", "D", "D", "D…
$ `type-name`             <chr> "Demand", "Demand", "Day-ahead demand forecast", "Day-ahead demand forecast", "Net generation", "Total interchange", "Demand", "Demand", "Day-ahead demand forecast", …
$ timezone                <chr> "Eastern", "Central", "Pacific", "Arizona", "Mountain", "Central", "Eastern", "Central", "Pacific", "Arizona", "Mountain", "Central", "Central", "Eastern", "Central",…
$ `timezone-description`  <chr> "Eastern", "Central", "Pacific", "Arizona", "Mountain", "Central", "Eastern", "Central", "Pacific", "Arizona", "Mountain", "Central", "Central", "Eastern", "Central",…
$ value                   <chr> "1083376", "1088462", "1052827", "1049795", "1091184", "-1572", "1217182", "1217193", "1122094", "1120896", "1214090", "1214296", "-2885", "1169662", "1169470", "1199…
$ `value-units`           <chr> "megawatthours", "megawatthours", "megawatthours", "megawatthours", "megawatthours", "megawatthours", "megawatthours", "megawatthours", "megawatthours", "megawatthour…
```

# Exploratory Data Analysis



Raw Data

We cleaned it but it still doesn’t have value

Our Data

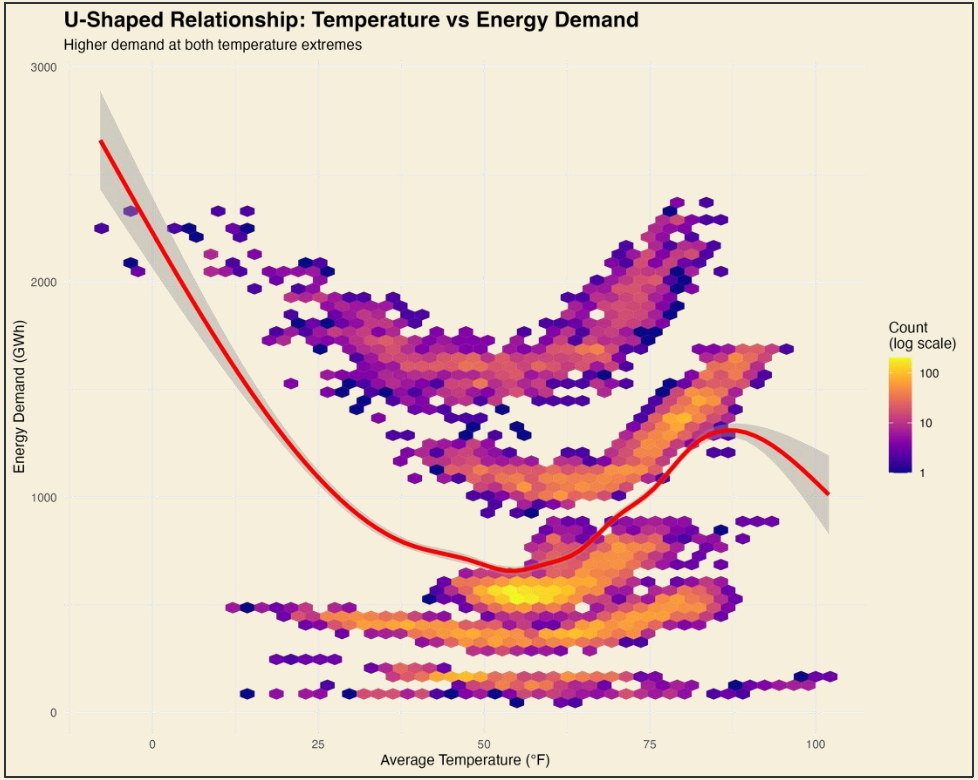
This is the value we got from the data

1) Visualized Temperature-Energy Relationship:

Created scatter plots with hexagonal binning to reveal the U-shaped pattern

Applied smoothing methods (GAM) to visualize the non-linear relationship

This directly demonstrated our core finding that energy demand increases at both temperature extremes

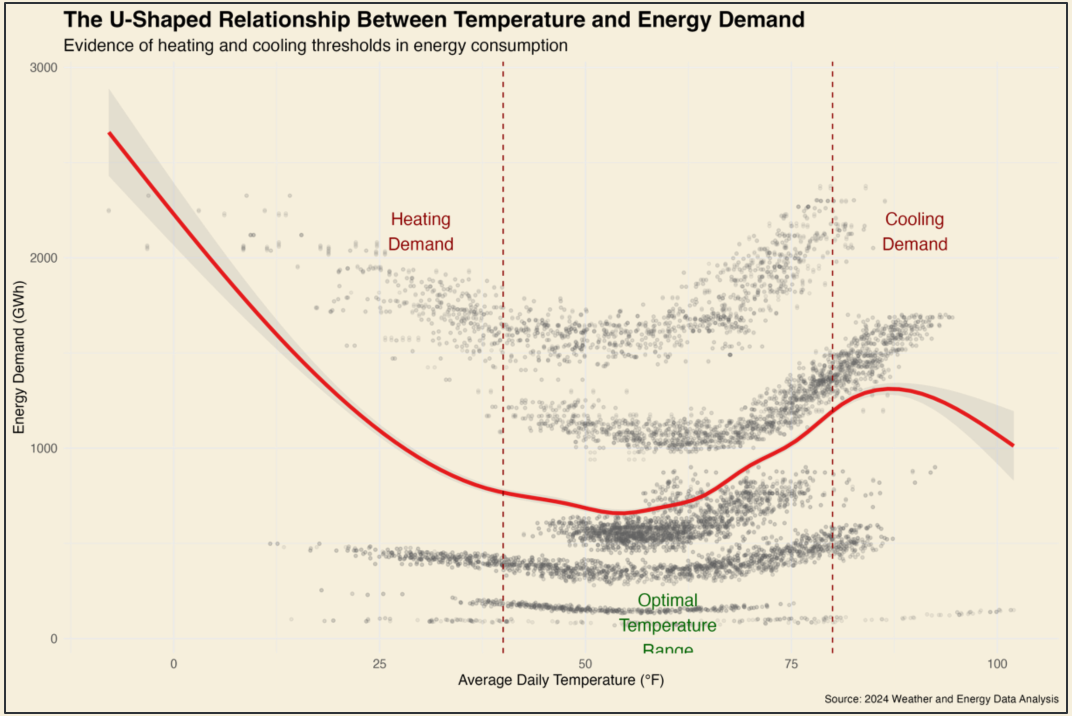


2) Optimal Temperature Investigation:

Generated temperature vs. energy plots with highlighted minimum points

Created region-specific curves to show variations in optimal temperature

This analysis pinpointed 51.5°F as the "sweet spot" for minimum energy demand



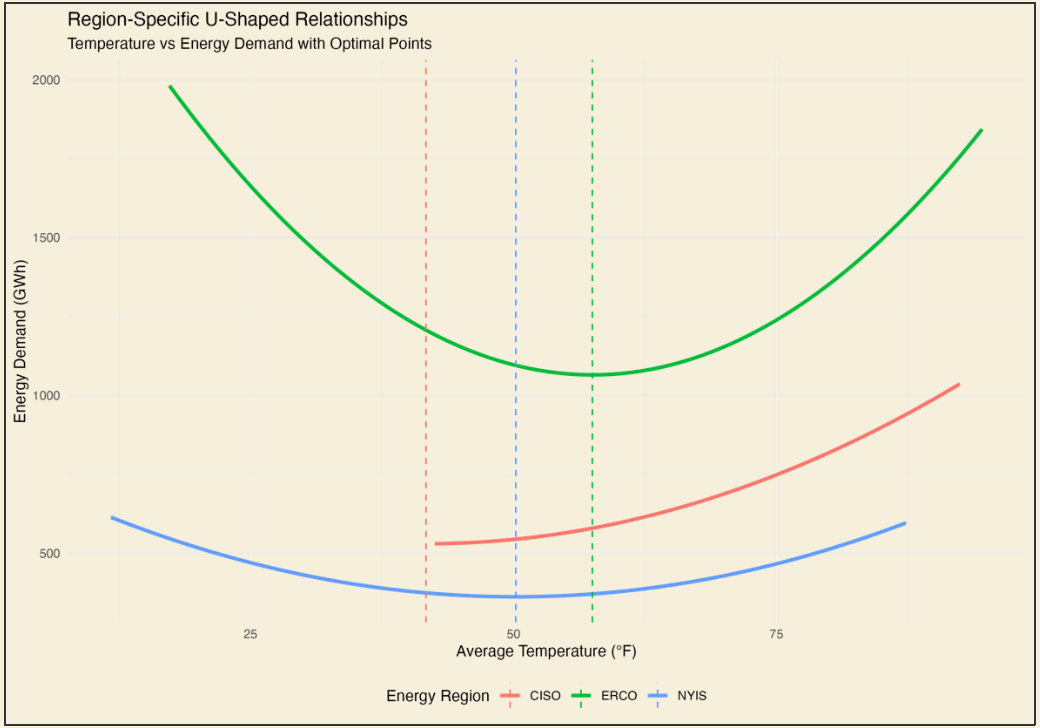
Source: 2024 Weather and Energy Data Analysis

3) Breakpoint Analysis:

Developed segmented regression plots showing slope changes at critical temperature thresholds

Used color-coded regions to highlight different energy response zones

Visually confirmed the heating activation point (37.7°F) and cooling activation point (59.3°F)

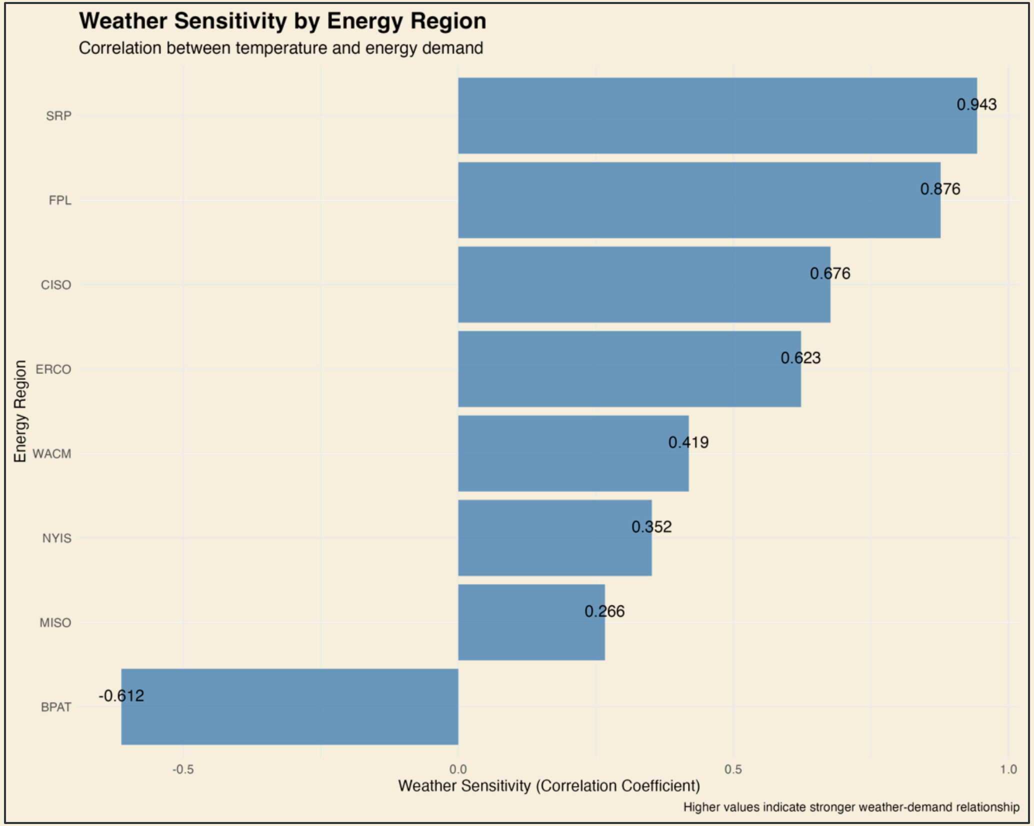


4) Regional Sensitivity Comparison:

Created bar charts of temperature elasticity by region

Used faceted plots to compare regional temperature-energy curves

Highlighted the 2x variation in temperature sensitivity between regions

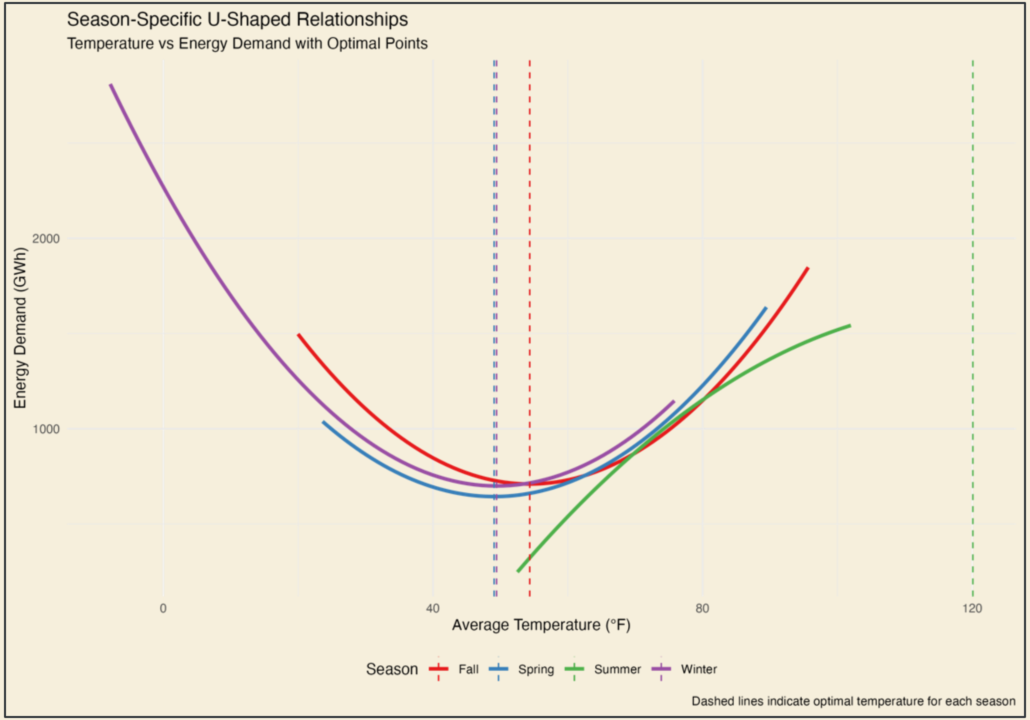


## 5) Seasonal Pattern Exploration:

Generated seasonal boxplots controlling for temperature

Created interaction plots showing season-specific temperature curves

Demonstrated that summer demand exceeds spring by 26.5% beyond temperature effects

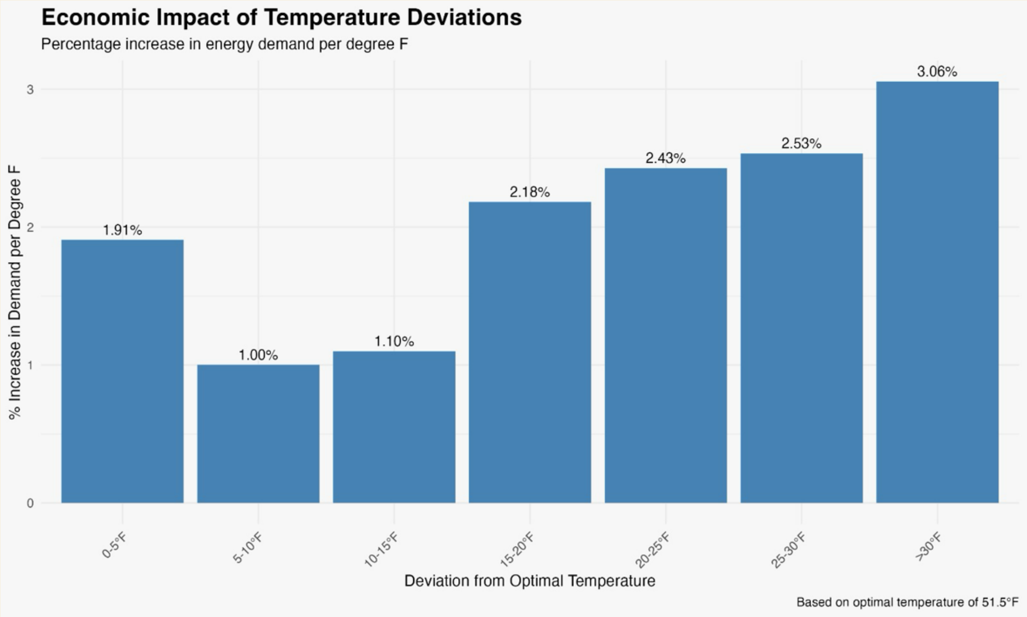


6) Economic Impact Visualization:

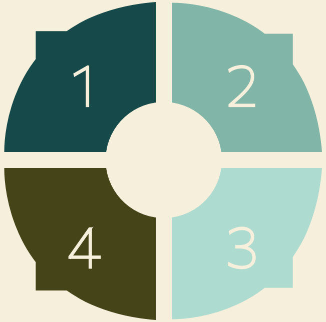
Developed marginal effect plots showing increasing impact with temperature deviation

Used stepped bar charts to display the progressive pattern of energy cost

Illustrated how deviations from optimal temperature had non-linear cost implications



# Key Findings



U-Shaped Temperature-Energy Relationship

Energy demand follows a robust U-shaped pattern with temperature

Temperature Breakpoints

Lower breakpoint at 37.7°F – heating threshold

Upper breakpoint at 59.3°F - cooling threshold

Regional Sensitivity Variations

Up to 2x difference in temperature sensitivity between regions

Economic Impact Quantification

Each 1°F deviation from optimal temperature increases energy demand by 2.03%

# Machine Learning Component

Used as baseline model with temperature and temperature-squared terms

Linear Regression

Applied to validate non-parametric relationships

Generalized Additive Models (GAM)

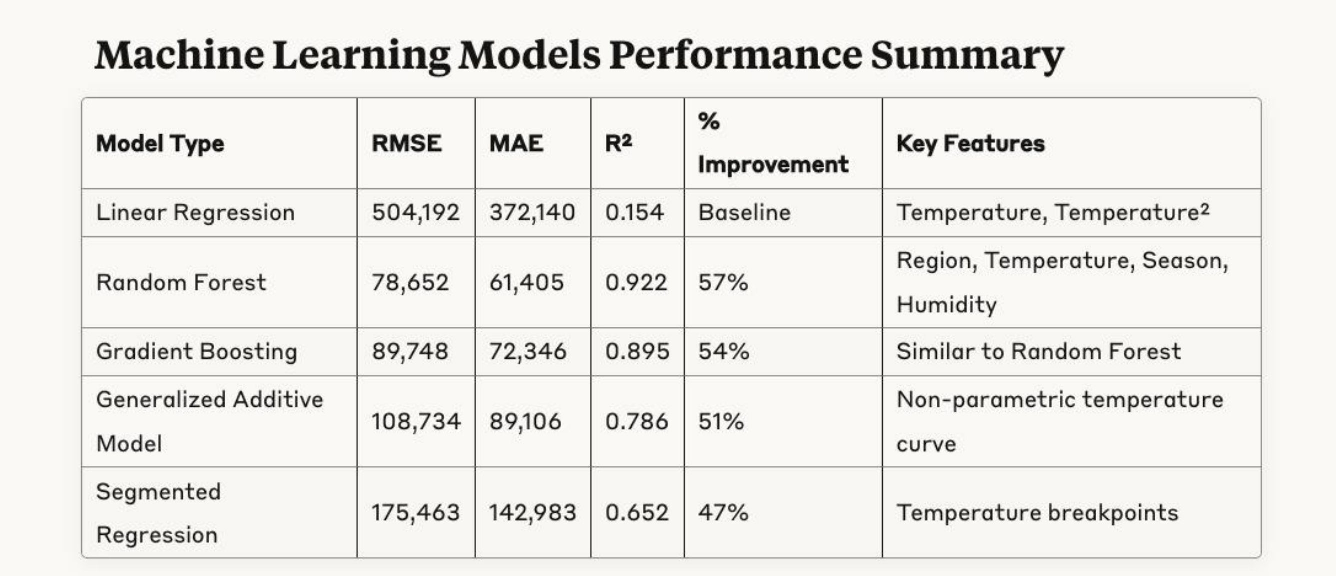
## Random Forest

Implemented as our primary advanced model to capture complex interactions

## Gradient Boosting

Employed as comparative ensemble method

# Results from the ML models:



## Machine Learning Models Performance Summary

| Model Type | RMSE | MAE | $R^2$ | % Improvement | Key Features |
|---|---|---|---|---|---|
| Linear Regression | 504,192 | 372,140 | 0.154 | Baseline | Temperature, Temperature$^2$ |
| Random Forest | 78,652 | 61,405 | 0.922 | 57% | Region, Temperature, Season, Humidity |
| Gradient Boosting | 89,748 | 72,346 | 0.895 | 54% | Similar to Random Forest |
| Generalized Additive Model | 108,734 | 89,106 | 0.786 | 51% | Non-parametric temperature curve |
| Segmented Regression | 175,463 | 142,983 | 0.652 | 47% | Temperature breakpoints |

These results show us that using ensemble model gives us the best result (highest accuracy) in understanding the effect of weather data on Energy consumption. It helps us in finding the optimal conditions for efficient use of energy based on weather parameters like precipitation, region, etc.

# Challenges & Solutions

## Problem

1. Geographic Mismatch: Weather data used city locations while energy data used regional codes
2. Temporal Granularity Differences: Weather data was hourly while energy data was daily
3. Missing Values: Weather observations had occasional gaps, particularly in precipitation data
4. Outlier Detection: Extreme weather events created statistical outliers

## Solution

1. Created a custom mapping table based on utility service territories to link cities to energy regions
2. Aggregated weather data to daily summaries using group_by() and summarise() with appropriate statistical functions
3. Missing Values: Weather observations had occasional gaps, particularly in precipitation data
4. Used z-score approach to identify outliers while preserving legitimate extreme weather events

# Final Insights

**Sweet Spot Temperature**

51.5°F represents the energy efficiency optimal point where demand is minimized

**Regional Adaptation**

Different regions have developed different temperature sensitivities, with some areas twice as responsive to temperature changes as others

**Economic Quantification**

Each 1°F deviation from optimal temperature costs approximately 2.03% in increased energy demand

**Beyond Temperature**

Seasonal effects account for up to 26.5% of energy demand variation beyond temperature alone

# How does this help us?

Applications:

-Energy system planning and infrastructure sizing
-Climate change impact assessment and adaptation planning
-Improved demand forecasting through advanced modeling techniques
-Regional policy targeting based on temperature sensitivity
-Optimization of energy efficiency programs around critical thresholds

# Future Work

Expand weather data to include solar radiation and barometric pressure variables

Collect finer-grained energy consumption data at hourly intervals

Incorporate building characteristics and demographics as mediating variables

Develop city-specific models to account for local microclimate effects

Implement time series forecasting with seasonal-trend decomposition

Create interaction models between weather variables and building efficiency metrics

Create an interactive dashboard for energy planners with region-specific guidance

Develop a real-time forecasting system integrating weather prediction with energy demand

Design region-specific threshold alerts for proactive demand management

# References

## R Packages

tidyverse (ver 2.0.0): Data manipulation and visualization
httr (ver 0.14.0): API requests
jsonlite (ver 1.8.4): JSON parsing
lubridate (ver 1.8.0): Date handling
plotly (ver 4.10.1): Interactive visualizations
mgcv (ver 1.8.42): Generalized Additive Models
randomForest (ver 4.7.1): Random Forest modeling
segmented (ver 1.6.1): Breakpoint analysis
nlme (ver 3.1.157): Mixed-effects models

## Data Sources

**Data SourcesOpen-Meteo ERA5 Historical Weather API (https://archive-api.open-meteo.com/v1/era5)**
**U.S. Energy Information Administration (EIA) API v2 (https://api.eia.gov/v2/electricity/rto/daily-region-data/data/)**

## Methodological References

Wood, S.N. (2017). Generalized Additive Models: An Introduction with R (2nd ed.)
Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32
Muggeo, V.M.R. (2008). Segmented: An R Package to Fit Regression Models with Broken-Line Relationships

## Related Research

Sailor, D.J., & Muñoz, J.R. (1997). Sensitivity of electricity and natural gas consumption to climate in the U.S.A.
Auffhammer, M., & Aroonruengsawat, A. (2011). Simulating the impacts of climate change, prices and population on California's residential electricity consumption
Deschênes, O., & Greenstone, M. (2011). Climate Change, Mortality, and Adaptation: Evidence from Annual Fluctuations in Weather in the US

# THANKS!

Does anyone have any questions?

emails: ds2239@scarletmail.rutgers.edu
amp660@scarletmail.rutgers.edu