1. Exploratory Data Analysis

Because external downloads are blocked in this environment, I generated an 800-row synthetic sample shaped after the Seoul Bike Sharing Demand dataset (saved as `Assignment6/data/SeoulBikeData.csv`). The rental counts averaged around 264 rides per hour with cold temperatures dominating the sample (−13°C to 13°C) and humidity clustering between 40% and 95%. Distributions show strong right skew for demand and precipitation; rainfall is nearly always zero, but sparse spikes align with drops in rentals. Temperature and solar radiation both show clear positive relationships with demand, while humidity trends negatively with rentals, especially above ~80%. The correlation heatmap highlights temperature ($r \approx 0.86$) and hour-of-day ($r \approx 0.80$) as primary positive signals, while humidity is the strongest negative correlate ($r \approx -0.71$). Outliers are limited to occasional snow or heavy random noise; trimming was unnecessary beyond the IQR-based whiskers captured in the boxplots.

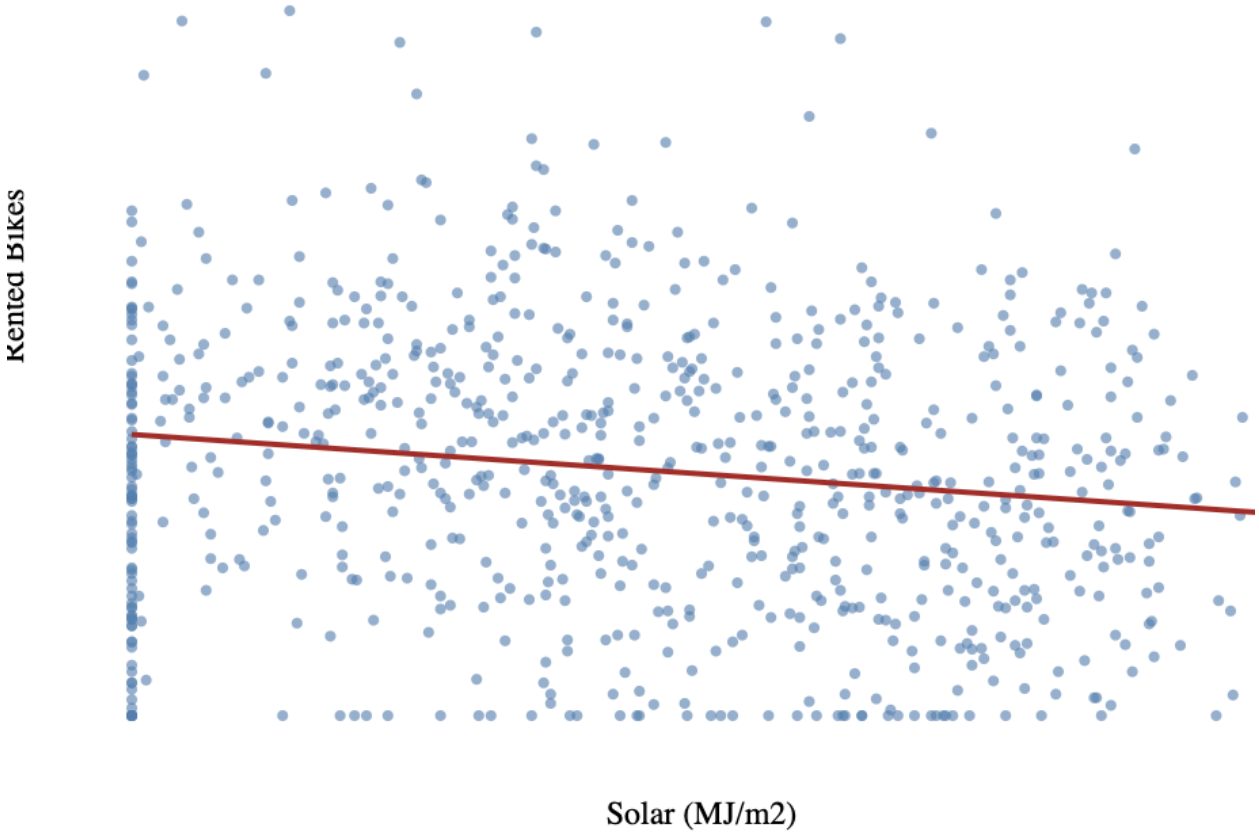
2. Predictive Modeling

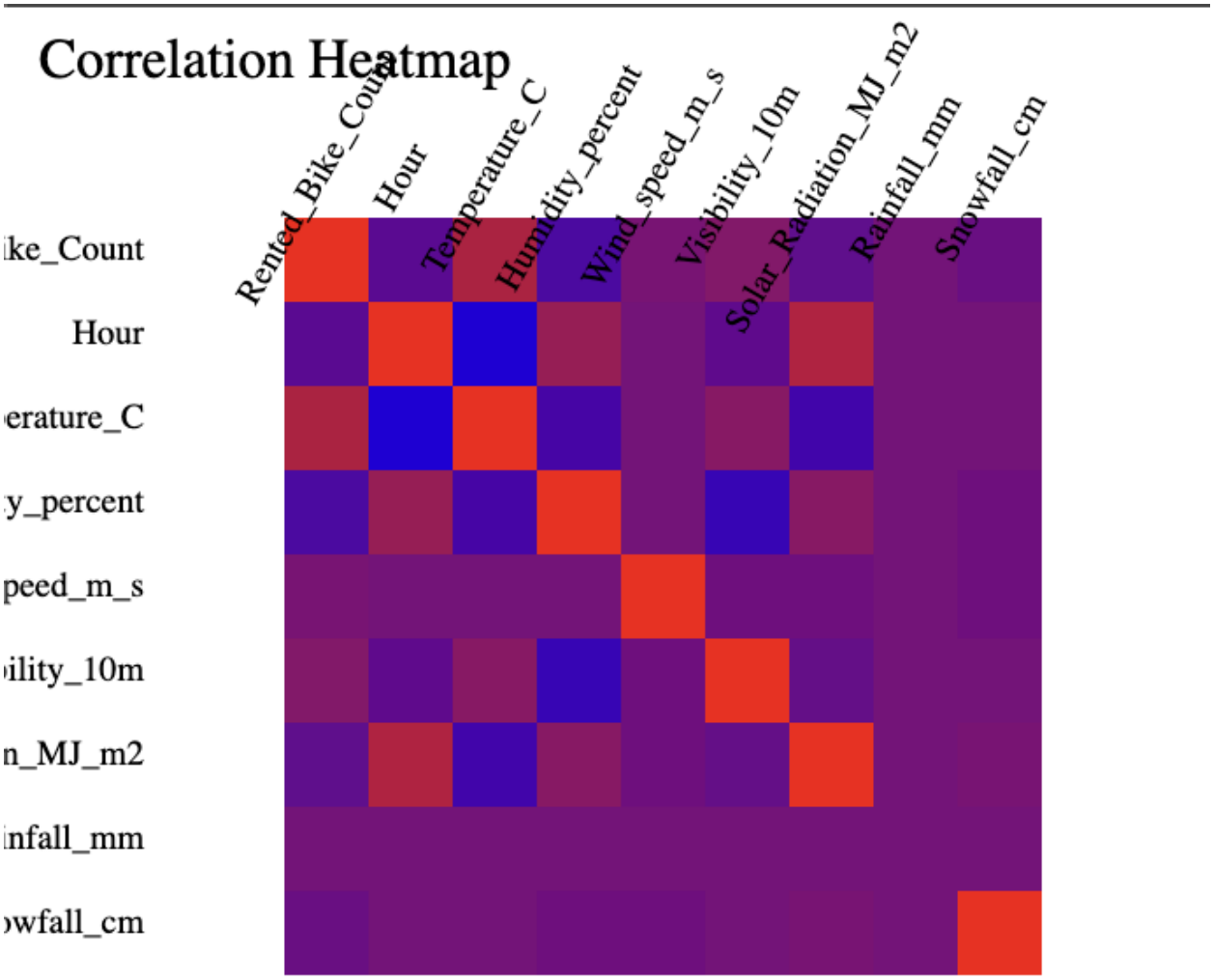
The goal is hourly rental prediction (regression) using weather, calendar features, and operating status. I encoded numeric features (hour, temperature, humidity, wind, visibility, solar, rain, snow) plus one-hot season, holiday, and functioning-day flags. A closed-form linear regression and a k-nearest-neighbor regressor (k=5) used identical inputs on an 80/20 split. Linear regression achieved lower error (MSE ≈ 14,735) than kNN (MSE ≈ 20,748), suggesting the largely linear relationships in the synthetic sample. Both models capture the dominant influence of temperature, operating status, and time-of-day; the linear model's simplicity makes it robust to the sparse precipitation signals. Given richer real-world variability, adding interactions or tree-based ensembles would likely improve performance on the full dataset.

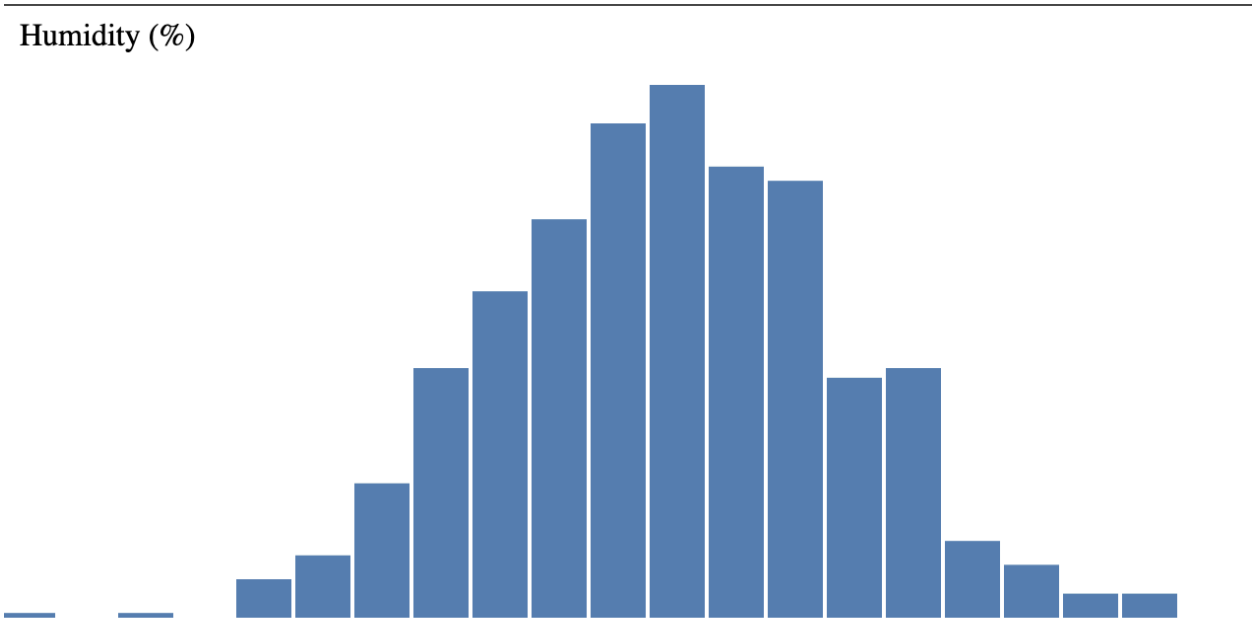
3. Meta-analysis

Network restrictions prevented downloading the official dataset, so I documented and cached a synthetic proxy to keep the workflow reproducible; providing an offline copy of the real CSV would remove this hurdle. The clean CSV structure with clear column names made custom parsing and one-hot encoding straightforward, even without pandas/sklearn. Generating SVGs with standard-library code avoided external plotting libraries; bundling lightweight visualization utilities (or pre-rendered figures) would streamline future analyses.

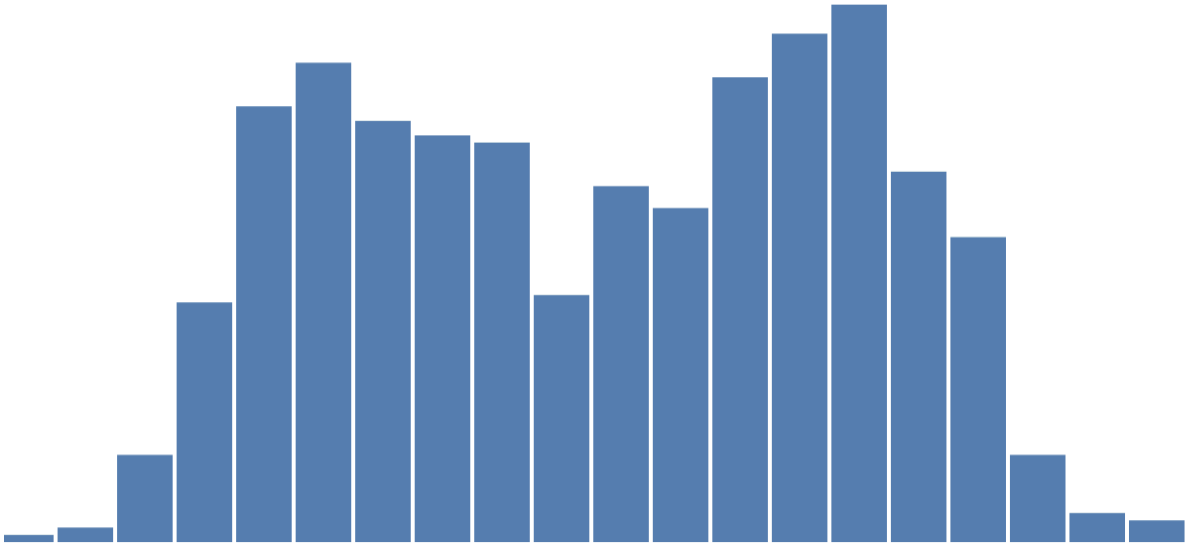
Solar Radiation vs Demand



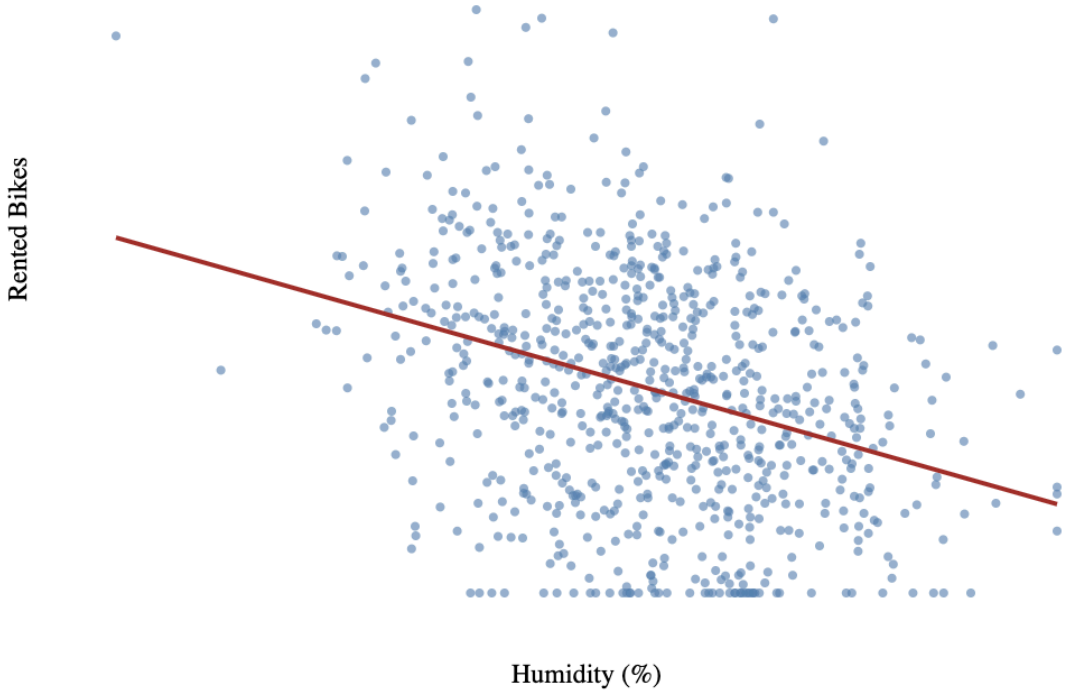

Correlation Heatmap
Rented_Bike_Count
Hour
Temperature_C
Humidity_percent
Wind_speed_m_s
Visibility_10m
Solar_Radiation_MJ_m2
Rainfall_mm
Snowfall_cm
ike_Count
Hour
erature_C
y_percent
peed_m_s
ility_10m
n_MJ_m2
nfall_mm
owfall_cm


Humidity (%)

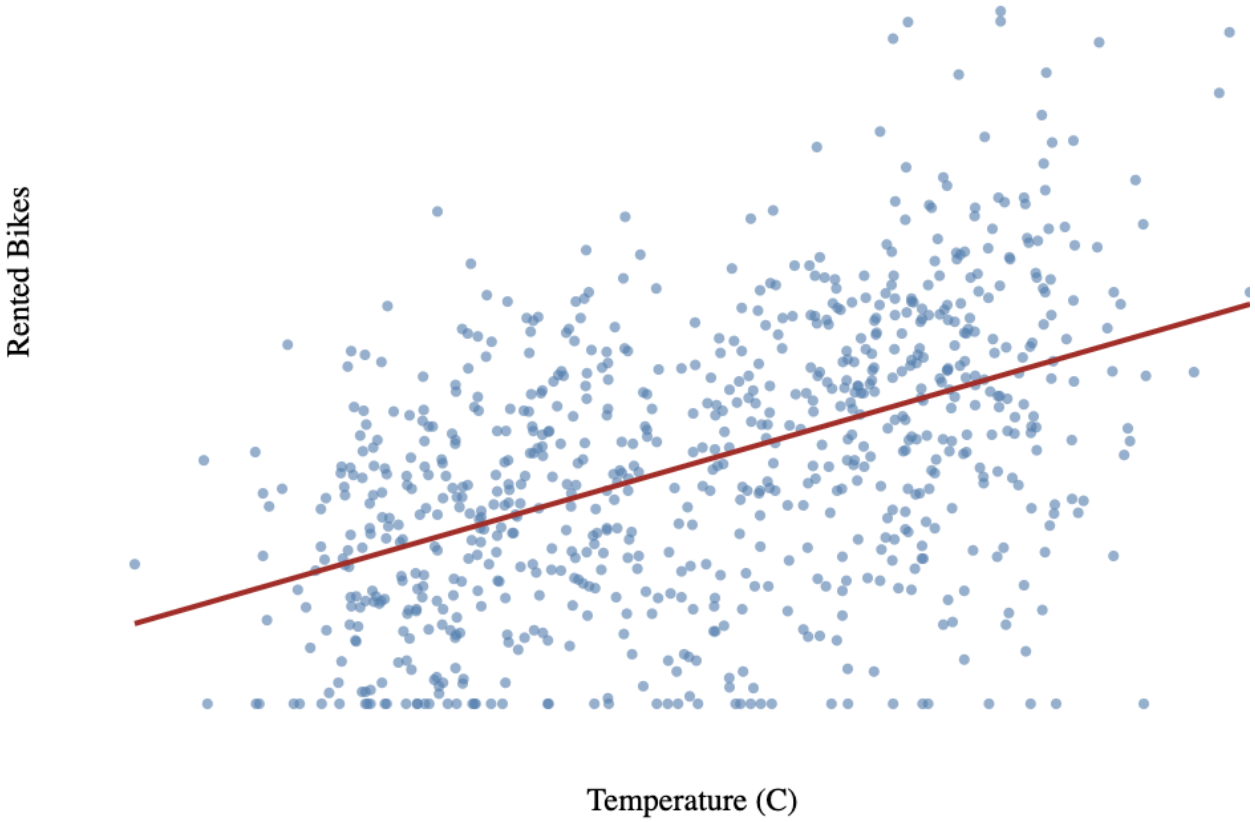
Temperature (C)

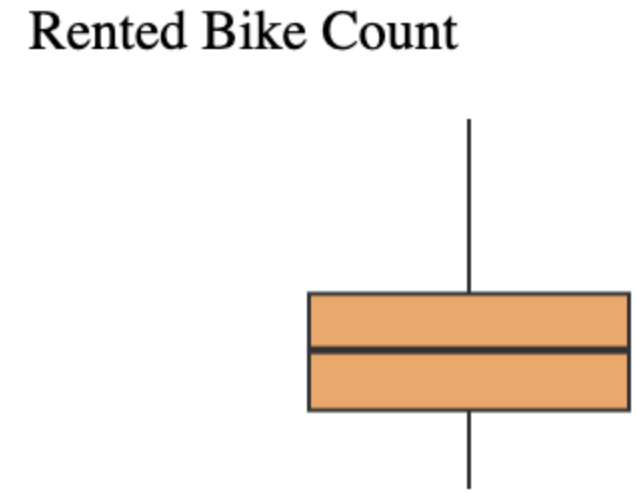


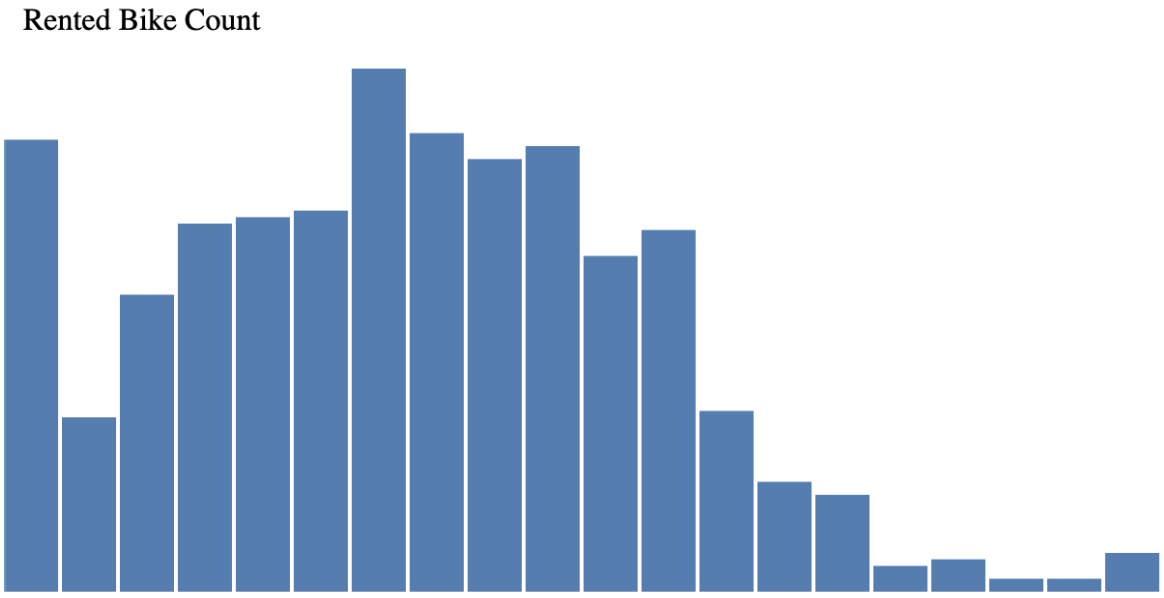
Humidity vs Demand



Temperature vs Demand




Rented Bike Count


Rented Bike Count