Data Analysis Final Assignment Report

Team: Transistor Titans
Zala Marušič, Volkan Sönmezler

1 Contributions

Clearly state each team member's specific contributions. Be concrete.

- Member 1: Volkan Sönmezler
  - Dataset selection and acquisition
  - Data quality analysis
  - Dimensionality Reduction
  - Statistical Theory Application
  - Report writing
- Member 2: Zala Marušič
  - Dataset selection and acquisition
  - Data Preprocessing and Visualization
  - Probability Analysis
  - Regression Analysis
  - Report writing and Presentation

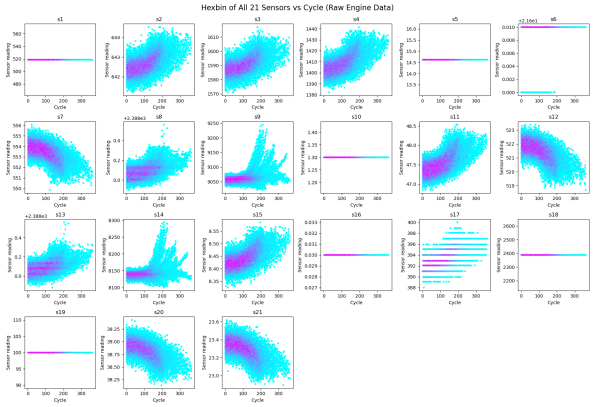
2 Dataset Description

- **Name and source:** NASA Predictive Maintenance (RUL) dataset, from Kaggle.
- **Suitability:** Sequential sensor data over engine cycles enables time-series analysis of degradation and remaining useful life.
- **Time span and frequency:** Recorded per engine cycle; each cycle is a discrete time step until failure.
- **Key variables:** Engine ID, Cycle, Operational settings (op1–op3), sensors (s1–s21), derived quantities (q1–q10), RUL.
- **Size and structure:** 20,631 rows, 26 raw features plus derived variables; target: RUL.
- **Missing data:** None; all readings complete.
- **Limitations:** Simulated data with limited operating conditions; some sensors show little variation; may not reflect real sensor noise or unexpected failures.

3 Task 1. Data Preprocessing and Basic Analysis

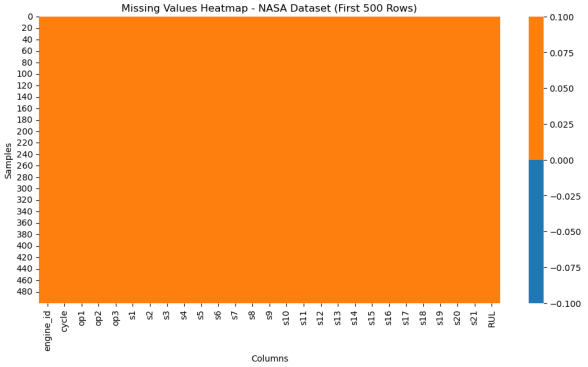
3.1 Basic statistical analysis using pandas

- Descriptive statistics (mean, std, min, max, quantiles) were computed for all sensors, settings, and RUL using pandas.
- Engine-level summaries were obtained by aggregating over cycles to compare degradation across engines.



3.2 Original data quality analysis including visualization

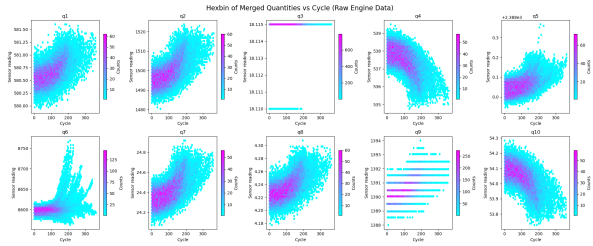
- Missingness was checked via value counts and a heatmap; no missing values were found in sensors, settings, engine IDs, or RUL.



- Outliers were examined via histograms, hexbin, and time-series plots.
- Consistency checks verified monotonically increasing cycles, no duplicates, and all sensor readings within plausible ranges.

3.3 Data preprocessing

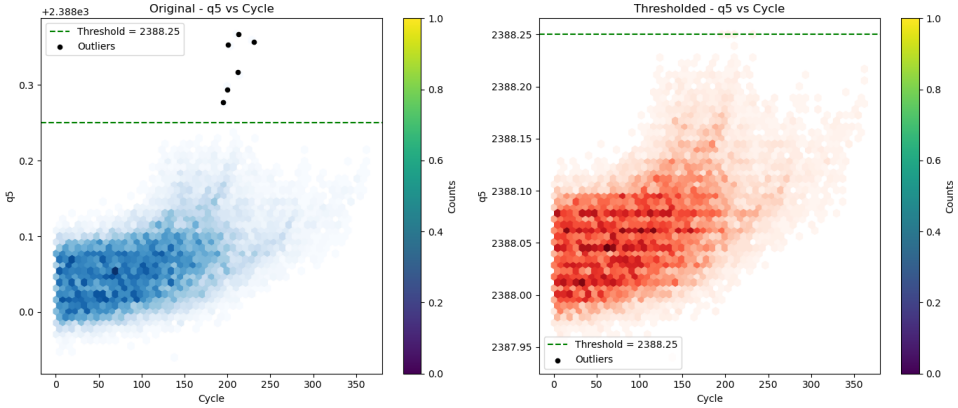
- Cleaning included computing RUL and aggregating the original sensor signals (s1–s21) into 10 derived quantities (q1–q10) based on similar numerical ranges, after which the raw sensor columns were removed to reduce dimensionality.

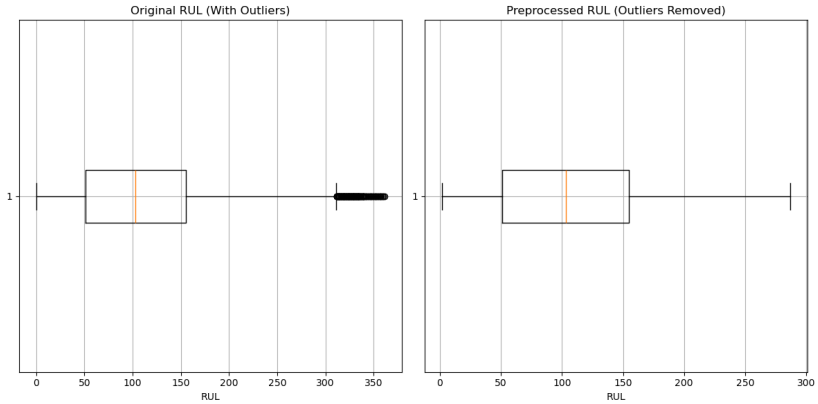


- No missing-value treatment was needed.
- Outliers were handled for q5 (clipped above 2388.25) and RUL (limited to 1st–99th percentiles).
- The final dataset contains engine IDs, cycle index, 10 derived quantities, and RUL.

3.4 Preprocessed vs original data visual analysis

- Hexbin and boxplots for q5 and RUL showed outlier removal while preserving overall data structure.



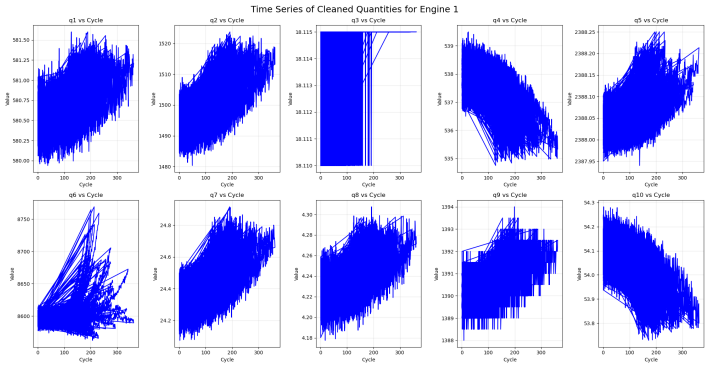


- Preprocessing reduced noise and clarified degradation trends, slightly compressing extreme upper-bound values.

4 Task 2. Visualization and Exploratory Analysis

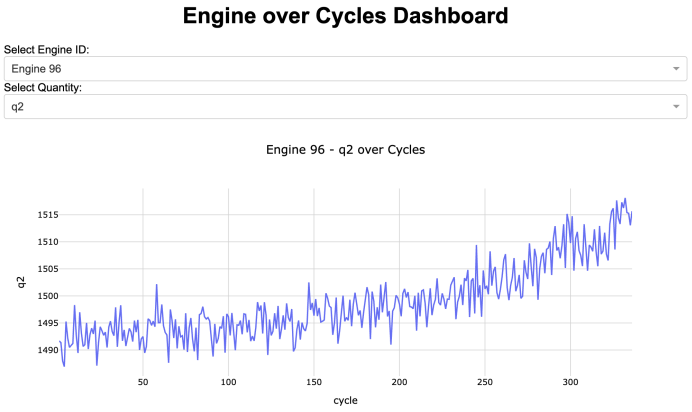
4.1 Time series visualizations

- Plot of main variable(s) over cycles:



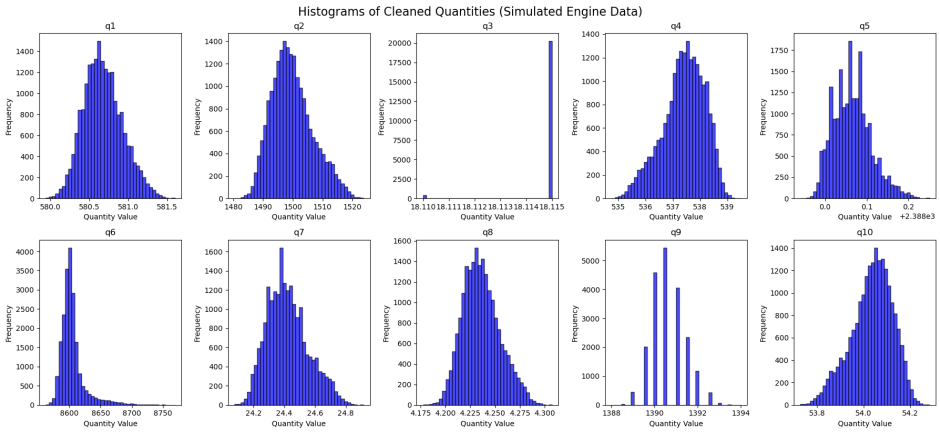
4.2 Interactive Dashboard Overview

- An interactive Jupyter dashboard was created to explore the 10 sensor-derived quantities (q1–q10) and RUL for all 100 engines.
- Users can select an Engine ID to view its time series and choose a quantity to plot over cycles.
- Purpose: enables fast visual comparison across many engines and quantities, which is cumbersome with static plots.



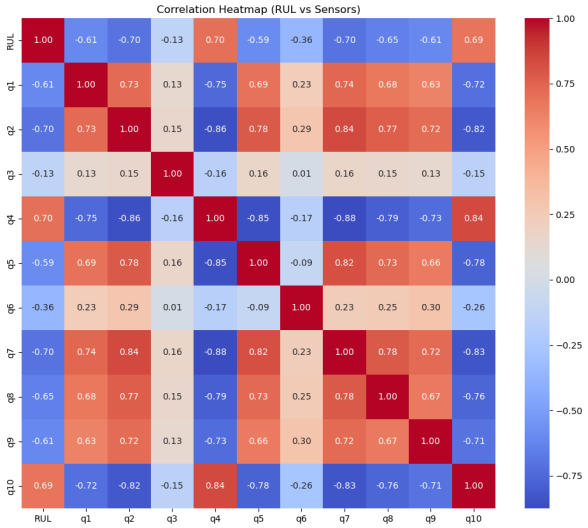
4.3 Distribution analysis with histograms

- Histograms for key numeric variables:



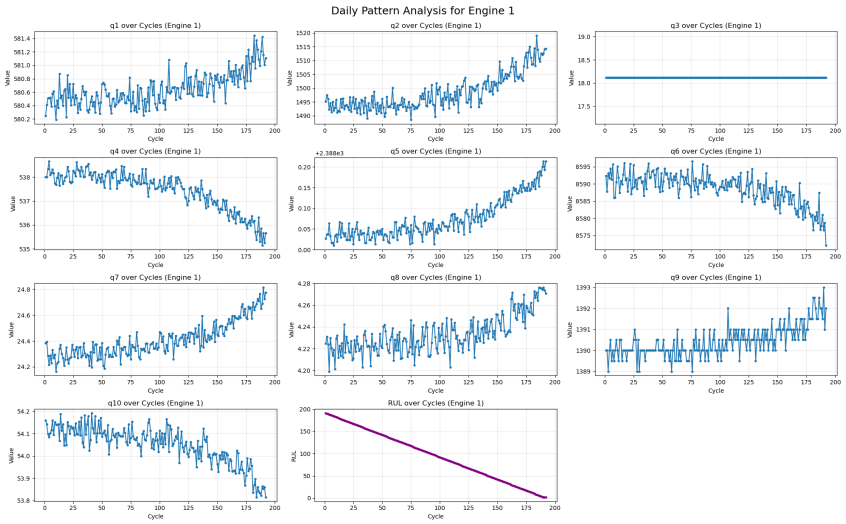
4.4 Correlation analysis and heatmaps

- **Correlation type used:** Pearson correlation measured linear relationships; Spearman was used as a robustness check for non-normal distributions and outliers.
- **Heatmap and top correlated pairs:** Strongest correlations were (RUL, q1), (RUL, q2), and (q1, q5), showing that q1, q2, and q5 capture the main degradation patterns.



4.5 Cycle pattern analysis

- Plots showing patterns over engine cycles (no timestamps available):



- Observed patterns:
  - The majority of quantities (q1, q2, q4, q7, q8, q10) exhibit stable, unimodal distributions with low variance, indicating consistent behavior across engines.
  - Quantity q3 shows extremely low variance and is nearly constant, providing little information for variability-based analysis.
  - Quantities q5 and q6 display higher variance and heavier tails, indicating residual noise.
  - Quantity q9 shows a discretized distribution with distinct spikes, likely due to limited sensor resolution.

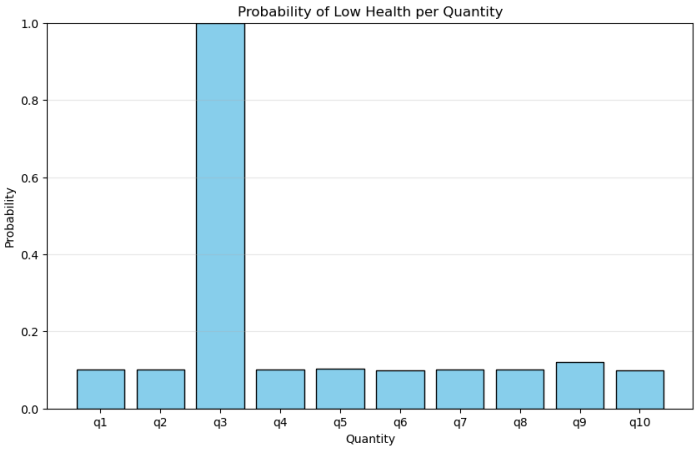
4.6 Summary of observed patterns

- Statement 1 (True): Most derived quantities exhibit stable, approximately unimodal distributions.
- Statement 2 (True): Quantity q3 provides limited information for variability-based analysis.
- Statement 3 (True): Some quantities remain noisy even after preprocessing.

5 Task 3. Probability Analysis

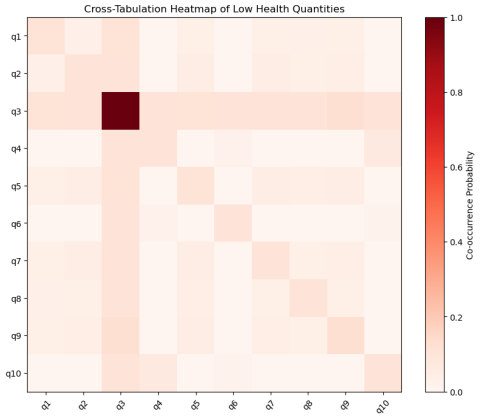
5.1 Threshold-based probability estimation

- Low-health thresholds were defined per derived quantity (q1–q10), producing binary low-health indicators.
- The probability of low health was estimated as the fraction of observations exceeding each threshold.
- A bar plot visualized low-health probabilities, enabling comparison across quantities.



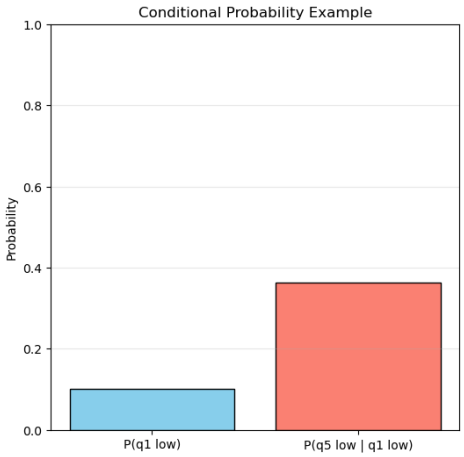
5.2 Cross tabulation analysis

- Low-health indicators were treated as categorical variables.
- A co-occurrence probability matrix was computed and visualized as a heatmap, revealing quantities that tend to degrade together, with q1 and q5 showing moderate joint degradation.



5.3 Conditional probability analysis

- Events were defined as specific quantities being in a low-health state.
- Marginal and conditional probabilities were estimated empirically from the low-health indicators.
- For example, $P(\text{q5 low} \mid \text{q1 low}) = 0.36$ shows that q5 is degraded in 36% of cases when q1 is degraded, exceeding q5's marginal probability and indicating dependency. In contrast, q3 is almost always low, yielding conditional probabilities near one and limited additional insight.



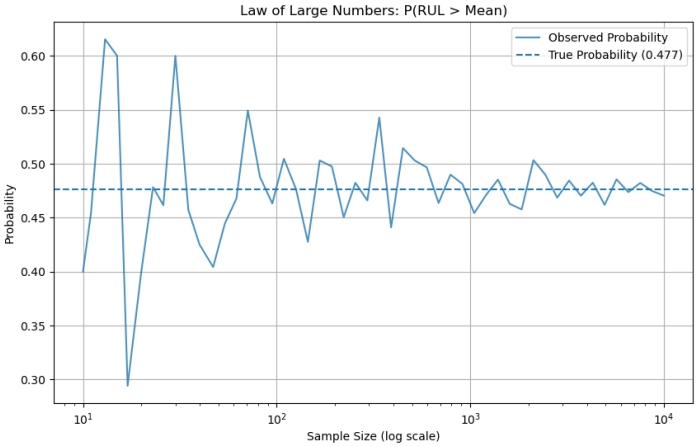
5.4 Summary of observations

- **Threshold probabilities:** Most quantities remain healthy, with a few showing higher degradation frequency.
- **Cross-tabulation:** Some quantities exhibit correlated degradation behavior.
- **Conditional probabilities:** Degradation likelihood can increase significantly when another quantity is already degraded.

6 Task 4. Statistical Theory Applications

6.1 Law of Large Numbers (LLN) demonstration

- Variable chosen and why it makes sense: RUL (*T24 – Total temperature at fan inlet*) was selected because it is a critical indicator of engine performance and health. The sensor exhibits natural variability over operational cycles, making it well suited to demonstrate the convergence of sample statistics to population parameters.
- Experiment: show sample mean as n increases: Random samples of increasing size ($n = 10, 50, 100, 500, 1000, 2000$) were drawn from the cleaned Sensor 2 dataset. For each sample size, the sample mean was calculated and plotted as a function of n.
- Plot and short interpretation: The plot shows the sample mean stabilising around the population mean (approximately 642.37) as n increases. Initial fluctuations observed for small sample sizes diminish as more observations are included, illustrating that larger samples yield more reliable estimates of the true mean.



6.2 Central Limit Theorem (CLT) application

- Sampling procedure (sample size, number of trials, with or without replacement): Samples were drawn with replacement from the Sensor 2 data using a fixed sample size of $n = 30$. A total of 1000 trials were performed to allow resampling from the finite dataset.
- Show distribution of sample means for increasing n: For sample sizes $n = 10, 30, 50, 100$, the distribution of sample means was analysed by computing 1000 sample means for each n and examining the resulting distributions.

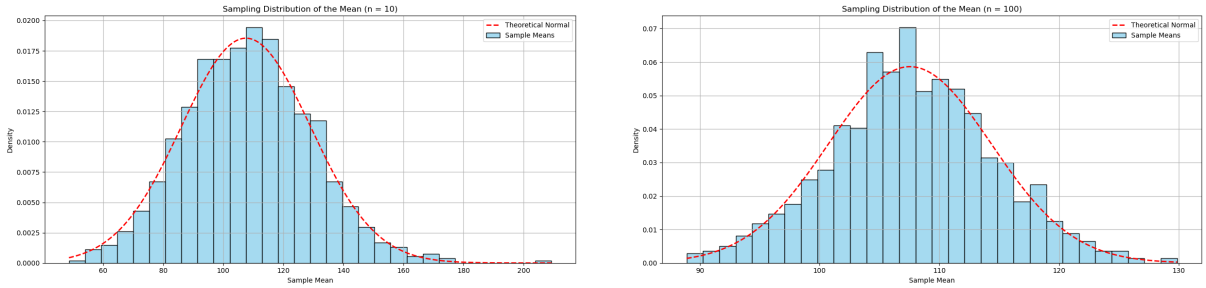


Figure 1: Law of Large Numbers demonstration for Sensor 2

Histograms of the sample means for $n = 10, 30, 50, 100$ were plotted and compared with the theoretical normal distribution, using the population mean and a standard deviation of $\sigma/\sqrt{n}$. For small sample sizes, the distributions appear skewed and irregular, while for larger n they become increasingly symmetric and bell-shaped, closely matching the normal curve.

6.3 Result interpretation

- Within the context of the NASA turbofan engine sensor data, the LLN demonstrates that as more sensor readings are collected, the average sensor value converges to the true population mean. This confirms that reliable estimates of engine parameters, such as inlet temperature, can be obtained despite inherent noise and variability in the data.
- The CLT analysis shows that the distribution of sample means approaches a normal distribution as sample size increases, enabling statistical inference such as confidence interval estimation for sensor thresholds. Deviations are observed for small sample sizes (e.g., $n = 10$), where skewness persists due to outliers and non-normality in the original sensor data, including extreme values arising from engine degradation cycles.

7 Task 5. Regression Analysis

7.1 Model Selection

- The target variable y was defined as the Remaining Useful Life (RUL).
- Regression models were trained using sensor measurements as predictors, including both multivariate linear regression (e.g., $q4$ and $q10$) and univariate polynomial regression (e.g., $q7$).
- An 80–20 train–test split with a fixed random seed was used to ensure reproducibility.

7.2 Model Fitting and Validation

- Linear regression models were fitted using ordinary least squares.
- Polynomial regression models of varying degree were evaluated to capture potential nonlinear degradation behavior.
- Standard 5-fold cross-validation was used to assess generalization performance.
- Model performance was evaluated using R^2 and RMSE on both training and test data.
- Residual plots and Q–Q plots were analyzed to identify systematic errors and deviations from normality.

7.3 Result Interpretation and Overall Conclusion

- Multivariate linear regression using $q4$ and $q10$ showed limited predictive power, particularly at high RUL values where sensor readings remain relatively constant.
- The models were most accurate when the actual RUL was low (approximately 0–50 cycles), indicating that predictions are more reliable close to failure.
- Polynomial regression confirmed a nonlinear relationship between sensor measurements and engine degradation, with higher sensor values corresponding to lower RUL.
- A third-degree polynomial provided the best trade-off between bias and variance; higher-degree models did not improve cross-validation performance and showed signs of overfitting.
- Increasing model complexity alone did not resolve prediction errors, highlighting the limitations imposed by early-life sensor plateaus and the need for more informative degradation features.

7.4 Detailed Error Analysis

- Coefficient analysis showed that sensors such as $q7$ are strong indicators of degradation, confirming their relevance for RUL estimation.
- The model tends to overshoot or undershoot RUL at specific ranges, reflecting the fact that engine degradation progresses slowly at first and accelerates rapidly near failure.
- Residual distributions exhibit heavier tails than a normal distribution, indicating the presence of extreme prediction errors.
- These deviations are likely caused by manufacturing variability between engines and sensor noise inherent in the C-MAPSS dataset.
- Polynomial models are particularly prone to extrapolation errors at the boundaries of the RUL range.

8 Key Findings and Conclusions

- **Preprocessing and EDA:**
  - RUL was computed per engine; sensors were grouped into q1–q10; outliers in q5 were clipped above 2388.25.
  - Hexbin plots, histograms, and time-series visualizations revealed clear cycle trends for several quantities; q3 behaved almost as a constant/binary signal.
  - PCA showed that PC1 was dominated by q1, q2, and q5 (64% variance), and the first two PCs explained 85%, indicating that 3–4 components captured most of the variance.
- **Probability analysis:**
  - Low-health flags were defined using 10th-percentile thresholds; most quantities were low in 10% of observations, q9 in 12%, and q3 in nearly 100%.
  - Co-occurrence heatmaps were generated, revealing moderate joint degradation, notably between q1 and q5.
  - Conditional probabilities were computed, showing dependencies; for example, $P(\text{q5 low} \mid \text{q1 low}) \approx 0.36$, exceeding q5's marginal probability. Low-support conditionals were interpreted cautiously.
- **LLN and CLT demonstrations:**
  - The Law of Large Numbers was demonstrated by showing that empirical probabilities converged to population values as sample size increased.
  - The Central Limit Theorem was illustrated by showing that the sampling distribution of means approached normality by n = 50–100.
- **Regression analysis:**
  - Polynomial regression models were fitted and improved training fit, but cross-validation revealed overfitting; residual plots showed non-random structure.
  - Using engine_id as a numeric predictor was found to be misleading; time-aware or per-engine validation was recommended.
  - PCA combined with regression indicated that q1, q2, and q5 were the most predictive of RUL, but extreme values and engines with unseen patterns were poorly predicted.
- **Limitations:**
  - The 10th-percentile threshold was arbitrary and influenced probability estimates.
  - Aggregation into q1–q10 masked per-sensor detail; q3's binary behavior limited its informativeness.
  - The cycle index was used as a proxy for time; some conditional probabilities had low support.
  - Using engine_id as a numeric predictor introduced potential confounding and data leakage.
- **Next steps if more time were available:**
  - Sensors could be analyzed individually to identify subtle early-warning patterns.
  - PCA or clustering could be applied over time to detect emerging degradation modes across engines, informing maintenance scheduling.

9 Reproducibility Notes

- Exact dataset source link: https://www.kaggle.com/datasets/behrad3d/nasa-cmaps
- Key libraries used:
  - **pandas** – data manipulation and analysis
  - **numpy** – numerical operations
  - **matplotlib.pyplot** – data visualization
  - **seaborn** – advanced visualizations
  - **scipy.stats** – statistical functions
  - **sklearn.model_selection** – train/test split, cross-validation
  - **sklearn.linear_model.LinearRegression** – linear regression modeling
  - **sklearn.preprocessing** – PolynomialFeatures, StandardScaler
  - **sklearn.metrics** – mean squared error, R^2 score
  - **sklearn.manifold.TSNE** – t-SNE embedding
  - **sklearn.decomposition.PCA** – PCA
  - **umap** – dimensionality reduction and visualization
  - **jupyter_dash (JupyterDash)** – inline Dash in notebooks
  - **dash** – app components (dcc, html, Input, Output)
  - **plotly.express** – interactive plotting in dashboard
- How to run the notebook end-to-end:
  - Ensure all required libraries are installed (see Key Libraries section).
  - Open the Jupyter notebook in a compatible environment (e.g., JupyterLab or VS Code).
  - Execute cells sequentially from top to bottom to preprocess data, analyze sensors, handle outliers, and train models.