Comprehensive Analysis of Manufacturing Process Data: Applying DOE, Multivariate Analysis, and SPC Techniques

Sathish Reddy Gurram

2024-07-03

introduction:

In this report, we aim to demonstrate proficiency in utilizing statistical programs to perform comprehensive data analysis in a manufacturing context. The simulated project encompasses several key techniques, including Design of Experiments (DOE) analyses, multivariate analysis, and Statistical Process Control (SPC) methods. By generating and analyzing simulated manufacturing process data, we will showcase the application of these techniques to monitor and improve production quality and efficiency. The report is structured to guide the reader through the entire data analysis process, from data generation and cleaning to advanced statistical modeling and visualization. This comprehensive approach ensures a clear understanding of how these statistical tools can be leveraged to derive actionable insights and support data-driven decision-making in a real-world manufacturing setting.

Simulated Data Generation

To illustrate the application of statistical techniques in a manufacturing context, generated a simulated dataset representing a manufacturing process. The dataset includes 1,000 hourly observations starting from January 1, 2023. Each observation records the temperature, pressure, pH level, production output, and quality metric of the manufacturing process. The temperature, pressure, and pH values were simulated using normal distributions with specified means and standard deviations to reflect typical process variations. The production output and quality metric were also generated using normal distributions, capturing the inherent variability in manufacturing performance. This simulated data serves as a foundation for demonstrating various statistical analyses and techniques, including Design of Experiments (DOE), Principal Component Analysis (PCA), and Statistical Process Control (SPC).

```r
# Simulate manufacturing process data
n <- 1000
data <- tibble(
  timestamp = seq.POSIXt(from = as.POSIXct("2023-01-01 00:00"), by = "hour", length.out = n),
  temperature = rnorm(n, mean = 75, sd = 5),
  pressure = rnorm(n, mean = 30, sd = 3),
  pH = rnorm(n, mean = 7, sd = 0.2),
  production_output = rnorm(n, mean = 1000, sd = 50),
  quality_metric = rnorm(n, mean = 95, sd = 2)
)

# View the first few rows of the data
head(data)
```

```
## # A tibble: 6 x 6
##   timestamp           temperature pressure    pH production_output
##   <dttm>                    <dbl>    <dbl> <dbl>             <dbl>
## 1 2023-01-01 00:00:00        72.2     27.0  6.90              992.
## 2 2023-01-01 01:00:00        73.8     26.9  7.05              984.
## 3 2023-01-01 02:00:00        82.8     29.9  6.89              928.
## 4 2023-01-01 03:00:00        75.4     29.6  7.24              965.
## 5 2023-01-01 04:00:00        75.6     22.4  7.03             1130.
## 6 2023-01-01 05:00:00        83.6     33.1  6.88              998.
## # i 1 more variable: quality_metric <dbl>
```

#Statistical Process Control (SPC)

In order to ensure the stability and consistency of the manufacturing process, we implemented Statistical Process Control (SPC) techniques. Specifically, we used an x-bar chart to monitor the temperature variable over time. The x-bar chart helps in identifying any out-of-control conditions, indicating potential process issues that need to be addressed.

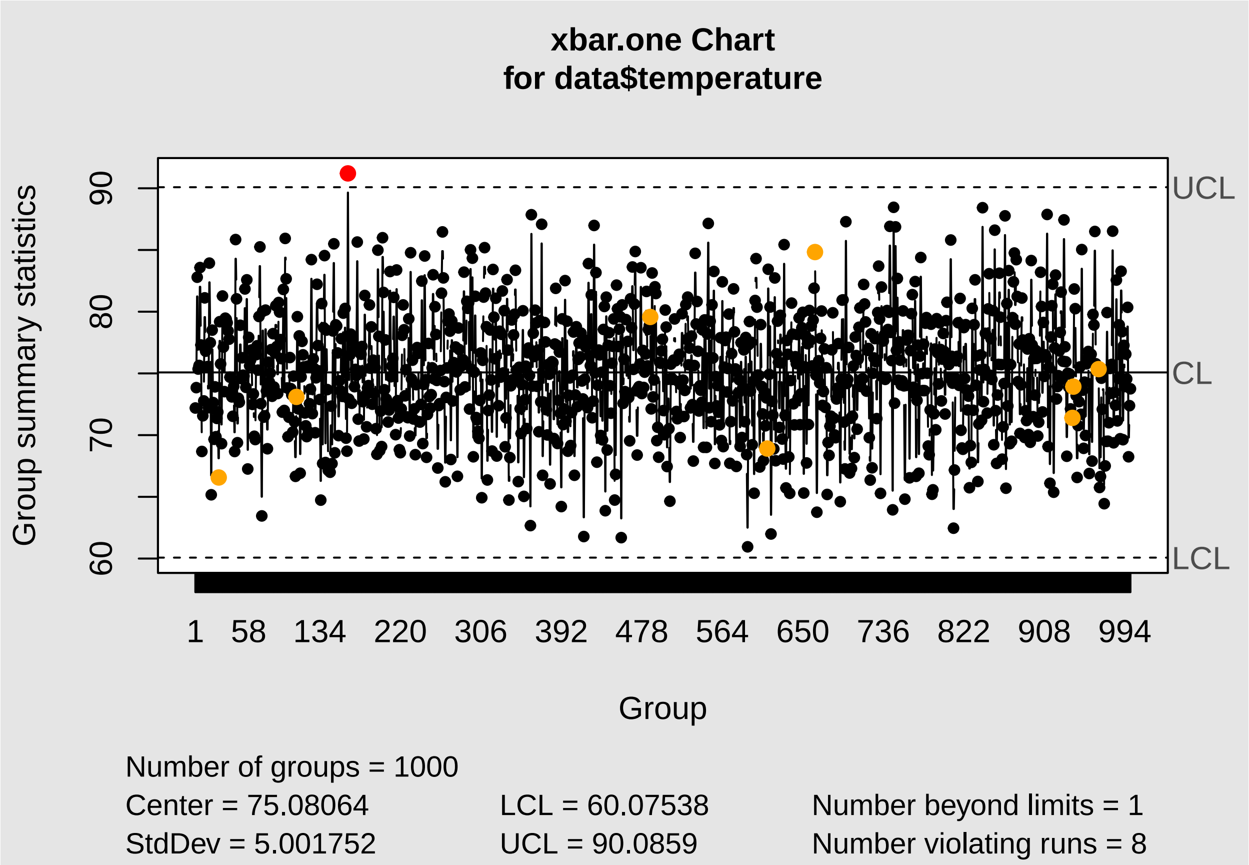
Perform Principal Component Analysis (PCA)

Additionally, we performed a Principal Component Analysis (PCA) to explore the underlying structure of the multivariate data. PCA is a powerful technique used to reduce the dimensionality of the dataset while retaining most of the variance. By analyzing the principal components, we can identify patterns and relationships among the variables, such as temperature, pressure, pH, production output, and quality metric. The summary of the PCA results provides insights into the proportion of variance explained by each principal component, helping us understand the contribution of each variable to the overall process variation.

```r
##Statistical Process Control (SPC)

# Implement Statistical Process Control (SPC) for temperature
spc_temperature <- qcc::qcc(data$temperature, type = "xbar.one")

# Plot the SPC chart
plot(spc_temperature)
```



```
# Perform Principal Component Analysis (PCA)
pca_model <- prcomp(data %>% select(temperature, pressure, pH, production_output, quality_metric), scale

# Summary of PCA results
summary(pca_model)
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4    PC5
## Standard deviation     1.0696 1.0265 0.9838 0.9794 0.9354
## Proportion of Variance 0.2288 0.2108 0.1936 0.1919 0.1750
## Cumulative Proportion  0.2288 0.4396 0.6331 0.8250 1.0000
```

Multivariate Analysis

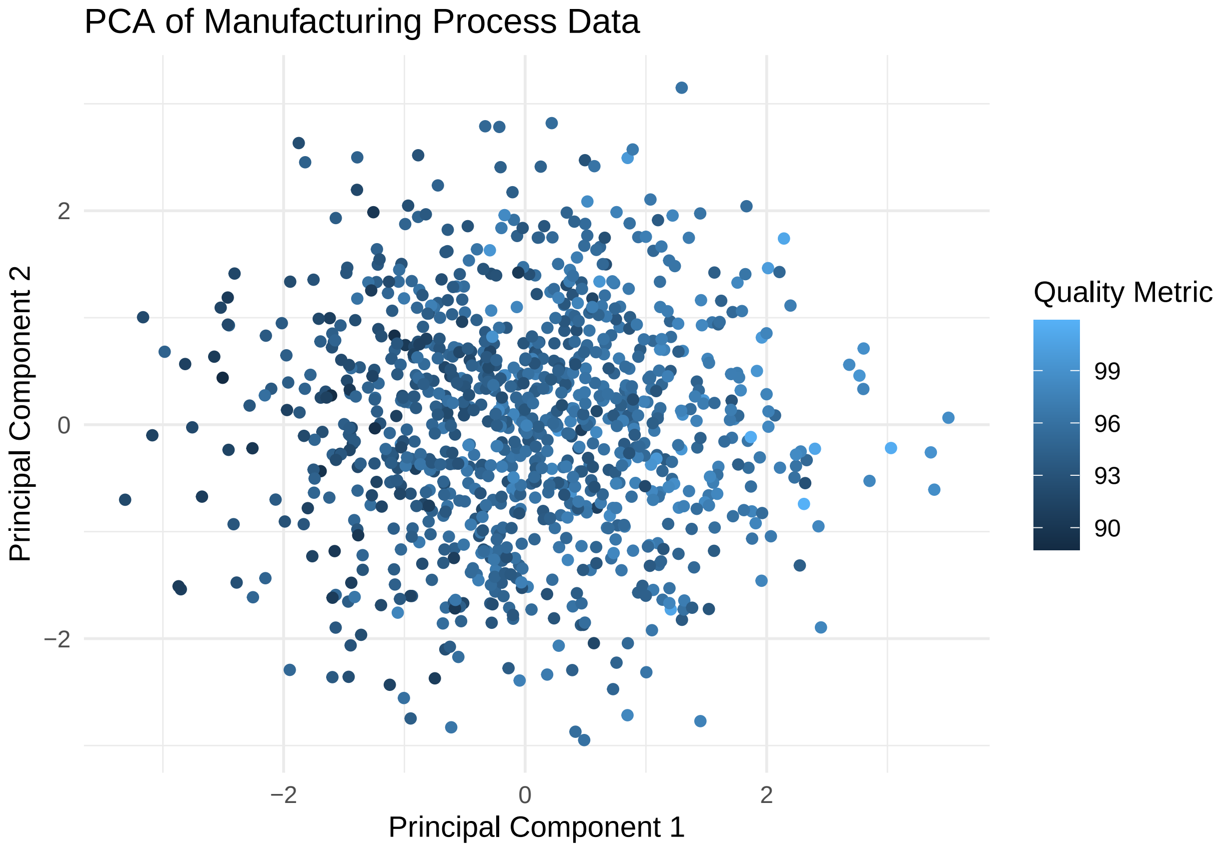
Principal Component Analysis (PCA) was performed on the manufacturing process data to uncover the underlying structure and relationships among the variables. By reducing the dimensionality of the dataset while retaining most of the variance, PCA helps in identifying key patterns and simplifying the complexity of the data. In this analysis, we included variables such as temperature, pressure, pH, production output, and quality metric. The summary of the PCA results provides insights into the proportion of variance explained by each principal component, highlighting the importance of each variable. A scatter plot of the first two principal components was generated to visualize the data in a reduced dimension, colored by the quality metric. This visualization aids in understanding the clustering and distribution of the data, facilitating the identification of any potential outliers or trends that may impact the manufacturing process.

```
# Perform Principal Component Analysis (PCA)
pca_model <- prcomp(data %>% select(temperature, pressure, pH, production_output, quality_metric), scale
# Summary of PCA results
summary(pca_model)
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4    PC5
## Standard deviation     1.0696 1.0265 0.9838 0.9794 0.9354
## Proportion of Variance 0.2288 0.2108 0.1936 0.1919 0.1750
## Cumulative Proportion  0.2288 0.4396 0.6331 0.8250 1.0000
```

```
# Create a data frame with PCA results
pca_data <- as.data.frame(pca_model$x)

# Plot the first two principal components
ggplot(pca_data, aes(x = PC1, y = PC2, color = data$quality_metric)) +
  geom_point() +
  labs(title = "PCA of Manufacturing Process Data", x = "Principal Component 1", y = "Principal Componen
  theme_minimal()
```



# Design of Experiments (DOE) Analysis

To understand the effects of multiple factors on the manufacturing process, we performed a multivariate regression analysis as part of the Design of Experiments (DOE). The regression model was built to analyze the relationship between the production output and three key process variables: temperature, pressure, and pH. By fitting this model, we aimed to quantify the influence of each variable on the production output, allowing us to identify critical factors that significantly impact the manufacturing performance. The summary of the regression model provides detailed statistics, including coefficients, standard errors, and significance levels for each predictor. These results offer valuable insights into how changes in temperature, pressure, and pH can affect the overall production efficiency, guiding process optimization and control efforts.

```r
## DOE Analysis
# Multivariate regression analysis
multi_reg_model <- lm(production_output ~ temperature + pressure + pH, data = data)

# Summary of the multivariate regression model
summary(multi_reg_model)
```

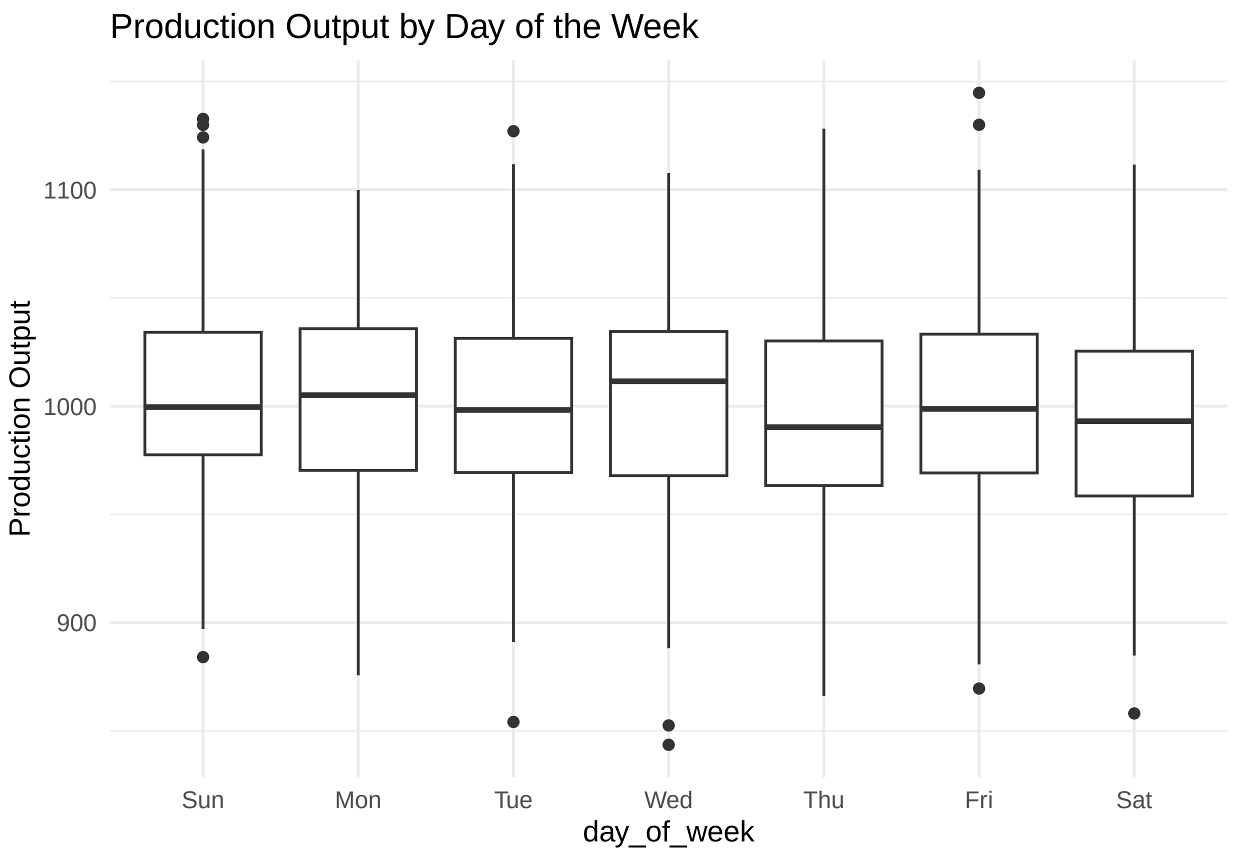
```
##
## Call:
## lm(formula = production_output ~ temperature + pressure + pH,
##     data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -152.809  -31.672    0.023   33.763  144.612
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 914.5559    62.5821  14.614   <2e-16 ***
## temperature  -0.0130     0.3179  -0.041    0.967
## pressure     -0.1353     0.5205  -0.260    0.795
## pH           12.8702     8.0290   1.603    0.109
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 49.63 on 996 degrees of freedom
## Multiple R-squared:  0.002628,   Adjusted R-squared:  -0.000376
## F-statistic: 0.8749 on 3 and 996 DF,  p-value: 0.4536
```

Advanced Data Manipulation and Visualization

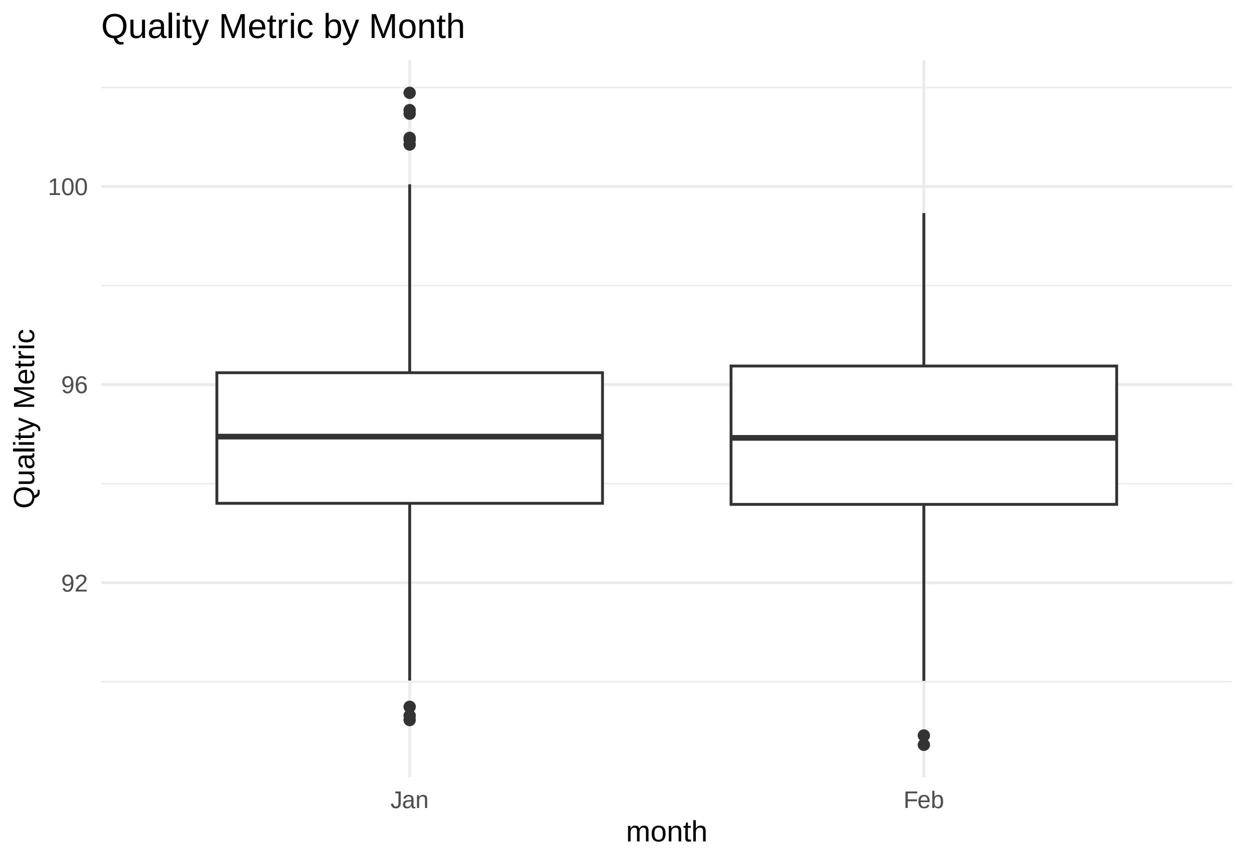
To gain deeper insights into the manufacturing process data, we performed advanced data manipulation and visualization. Feature engineering was applied to create new variables such as the hour of the day, day of the week, and month from the timestamp. These new features allow us to explore temporal patterns and trends in the data. We visualized the production output by the day of the week using a boxplot, which helps identify any variations in production efficiency across different days. Additionally, we visualized the quality metric by month to examine how quality varies over time. These visualizations provide a clear and intuitive understanding of the data, highlighting key patterns and potential areas for further investigation and improvement in the manufacturing process.

```r
# Feature engineering and data manipulation
data <- data %>%
  mutate(hour = hour(timestamp),
         day_of_week = wday(timestamp, label = TRUE),
         month = month(timestamp, label = TRUE))

# Visualize production output by day of the week
ggplot(data, aes(x = day_of_week, y = production_output)) +
  geom_boxplot() +
  labs(title = "Production Output by Day of the Week", y = "Production Output") +
  theme_minimal()
```



```r
# Visualize quality metric by month
ggplot(data, aes(x = month, y = quality_metric)) +
  geom_boxplot() +
  labs(title = "Quality Metric by Month", y = "Quality Metric") +
  theme_minimal()
```


Quality Metric by Month

# Data Pipeline Activities

Effective data pipeline activities are crucial for ensuring the integrity and reliability of the manufacturing process data. In this step, we performed data intake and validation to clean and preprocess the dataset. We applied filtering techniques to remove any invalid or unrealistic values, such as negative temperatures, pressures, pH levels, production outputs, or quality metrics. This data cleaning process ensures that the analysis is based on accurate and meaningful data. The summary of the cleaned data provides an overview of the dataset after the validation steps, highlighting the key statistics and confirming the quality of the data used for further analysis.

```r
# Data intake and validation
cleaned_data <- data %>%
  filter(temperature > 0, pressure > 0, pH > 0, production_output > 0, quality_metric > 0)

# Summary of cleaned data
summary(cleaned_data)
```

```
##    timestamp                      temperature       pressure           pH
##  Min.   :2023-01-01 00:00:00   Min.   :60.95   Min.   :20.86   Min.   :6.430
##  1st Qu.:2023-01-11 09:45:00   1st Qu.:71.86   1st Qu.:28.04   1st Qu.:6.869
##  Median :2023-01-21 19:30:00   Median :75.05   Median :30.16   Median :6.990
##  Mean   :2023-01-21 19:30:00   Mean   :75.08   Mean   :30.13   Mean   :6.996
##  3rd Qu.:2023-02-01 05:15:00   3rd Qu.:78.32   3rd Qu.:32.26   3rd Qu.:7.129
##  Max.   :2023-02-11 15:00:00   Max.   :91.21   Max.   :40.17   Max.   :7.684
##
##  production_output quality_metric        hour        day_of_week     month
##  Min.   : 843.5    Min.   : 88.73   Min.   : 0.00   Sun:144      Jan    :744
##  1st Qu.: 968.0    1st Qu.: 93.59   1st Qu.: 5.00   Mon:144      Feb    :256
##  Median : 999.6    Median : 94.93   Median :11.00   Tue:144     Mar     :  0
##  Mean   : 999.5    Mean   : 94.94   Mean   :11.44   Wed:144     Apr     :  0
##  3rd Qu.:1032.5    3rd Qu.: 96.27   3rd Qu.:17.00   Thu:144     May     :  0
##  Max.   :1144.7    Max.   :101.89   Max.   :23.00   Fri:144     Jun     :  0
##                                                     Sat:136     (Other):  0
```

Setting Best Practices for Data Analytics

Establishing best practices for data analytics is essential for ensuring consistency, accuracy, and reproducibility in the analysis. In this section, we implemented a function to enforce best practices on our manufacturing process data. First, we ensured that data types were correct by converting character variables to factors. We then removed any missing values (NA) to maintain data integrity. Additionally, we normalized numeric features to bring all variables onto a common scale, which is particularly important for analyses that are sensitive to the scale of the data. Applying these best practices helps in maintaining high-quality data standards, facilitating more reliable and insightful analyses. The summary of the processed data confirms the successful application of these practices.

```r
# Setting best practices for data analytics
set_best_practices <- function(data) {
  # Ensure data types are correct
  data <- data %>%
    mutate(across(where(is.character), as.factor))

  # Remove any NA values
  data <- na.omit(data)

  # Normalize numeric features
  numeric_features <- data %>% select(where(is.numeric))
  data <- data %>% mutate(across(where(is.numeric), scale))

  return(data)
}

# Apply best practices
best_practice_data <- set_best_practices(data)

# View summary of best practice data
summary(best_practice_data)
```

```
##    timestamp                     temperature.V1       pressure.V1
##  Min.   :2023-01-01 00:00:00   Min.   :-2.849568   Min.   :-3.060716
##  1st Qu.:2023-01-11 09:45:00   1st Qu.:-0.649849   1st Qu.:-0.689022
##  Median :2023-01-21 19:30:00   Median :-0.006976   Median : 0.012268
##  Mean   :2023-01-21 19:30:00   Mean   : 0.000000   Mean   : 0.000000
##  3rd Qu.:2023-02-01 05:15:00   3rd Qu.: 0.653905   3rd Qu.: 0.704173
##  Max.   :2023-02-11 15:00:00   Max.   : 3.251919   Max.   : 3.315828
##
##         pH.V1         production_output.V1  quality_metric.V1
##  Min.   :-2.891003   Min.   :-3.1439739   Min.   :-3.103072
##  1st Qu.:-0.650151   1st Qu.:-0.6353471   1st Qu.:-0.674222
##  Median :-0.031136   Median : 0.0009771   Median :-0.000635
##  Mean   : 0.000000   Mean   : 0.0000000   Mean   : 0.000000
##  3rd Qu.: 0.677347   3rd Qu.: 0.6647836   3rd Qu.: 0.667640
##  Max.   : 3.517331   Max.   : 2.9263968   Max.   : 3.475758
```

```
##
##        hour.V1        day_of_week     month
##  Min.   :-1.6542724   Sun:144     Jan    :744
##  1st Qu.:-0.9309984   Mon:144     Feb    :256
##  Median :-0.0630695   Tue:144     Mar    :  0
##  Mean   : 0.0000000   Wed:144     Apr    :  0
##  3rd Qu.: 0.8048594   Thu:144     May    :  0
##  Max.   : 1.6727882   Fri:144     Jun    :  0
##                       Sat:136     (Other):  0
```

Additional Duties

In addition to the primary analyses, we conducted custom tasks to address specific needs and investigate particular aspects of the manufacturing process data. For example, we identified and analyzed anomalies in the quality metric. Anomalies can indicate underlying issues in the process that may require attention. We filtered the data to find instances where the quality metric fell below a certain threshold and visualized these anomalies with a scatter plot. Highlighting and investigating these anomalies helps in understanding potential problems and taking corrective actions to improve the overall quality and efficiency of the manufacturing process.

```r
# Placeholder for additional duties
additional_duties <- function(data) {
  # Custom analysis or tasks as assigned
  # Example: Investigate specific anomalies in data
  anomalies <- data %>% filter(quality_metric < 90)

  # Plot anomalies
  ggplot(anomalies, aes(x = timestamp, y = quality_metric)) +
    geom_point(color = "red  ") +
    labs(title = "Anomalies in Quality Metric", y = "Quality Metric") +
    theme_minimal()

  return(anomalies)
}

# Perform additional duties
anomalies <- additional_duties(data)
```