Exploratory Data Analysis

In this first level of data exploration, I will analyze and describe key aspects of the dataset, focusing on uncovering insights from my perspective. This level is designed for beginners, but I aim to provide a structured and thoughtful approach to answering the given questions. Specifically, I will explore the following:

- What is the first and last date readings were taken on?
- What is the average torque?
- Which assembly line has the highest readings of machine downtime?

Through this process, I will apply fundamental data analysis techniques to extract meaningful insights and set the foundation for deeper exploration in subsequent levels.

In [1]:

```python
# import the necessary libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings

from scipy.stats import sem
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_percentage_error as MAPE
from statsmodels.tsa.statespace.sarimax import SARIMAX
import scipy.stats as stats
from scipy.stats import chi2_contingency

from statsmodels.tsa.seasonal import STL
from predictr import Analysis

from typing import Tuple, Union
from itertools import product
from pandas.plotting import register_matplotlib_converters

#import reliability.Fitters as rf

register_matplotlib_converters()
%matplotlib inline
warnings.filterwarnings('ignore')
plt.style.use('seaborn-v0_8-colorblind')
plt.rcParams['figure.figsize'] = (16,9)
plt.rcParams['axes.labelsize'] = 16
plt.rcParams['axes.titlesize'] = 18
plt.rcParams['legend.fontsize'] = 14
plt.rcParams['xtick.labelsize'] = 14
plt.rcParams['ytick.labelsize'] = 14
```

In [2]:

```
# read the cleaned data as a dataframe
machine_ori = pd.read_csv('../data/machine_downtime_cleaned.csv', parse_da
machine_ori.columns
```

Out[2]:

```
Index(['Date', 'Machine_ID', 'Assembly_Line_No', 'Coolant_Temperature',
       'Hydraulic_Oil_Temperature', 'Spindle_Bearing_Temperature',
       'Spindle_Vibration', 'Tool_Vibration', 'Voltage(volts)', 'Torque
(Nm)',
       'Downtime', 'Hydraulic_Pressure(Pa)', 'Coolant_Pressure(Pa)',
       'Air_System_Pressure(Pa)', 'Cutting(N)', 'Spindle_Speed(RPS)'],
      dtype='object')
```

In [3]:

```
machine_ori.Date.dtype
```

Out[3]:

```
dtype('<M8[ns]')
```

Feature Engineering

- Extract month, day and year from the date column
- set date column as index

In [4]:

```
# extract year, month, day of the week and week
# from date column
machine_ori['Day'] = machine_ori.Date.dt.day_name()
machine_ori['Month'] = machine_ori.Date.dt.month
machine_ori['Year'] = machine_ori.Date.dt.year
machine_ori['Month_Name'] = machine_ori.Date.dt.month_name()
#machine_ori['Week'] = machine_ori.Date.dt.isocalendar().week.astype('int


# verify
machine_ori.head(3)
```

Out[4]:

	Date	Machine_ID	Assembly_Line_No	Coolant_Temperature	Hydraulic_Oil_Temperature
0	2021-12-08	Makino-L2-Unit1-2015	Shopfloor-L2	4.5	47.9
1	2021-12-17	Makino-L2-Unit1-2015	Shopfloor-L2	21.7	47.5
2	2021-12-17	Makino-L1-Unit1-2013	Shopfloor-L1	5.2	49.4

1.1 What is the first and last date readings were taken on?

The data from the machine downtime for the company was taken between 11th of November 2021 to 3rd of July 2024. So roughly, we can say that we have about 9 months of machine downtime data

In [5]:

```
#get the first and last date reading
machine_ori['Date'].agg(['min', 'max'])
```

Out[5]:

```
min   2021-12-08
max   2022-07-03
Name: Date, dtype: datetime64[ns]
```

1.2 What is the average torque?

In this level of data exploration, one of the key tasks is to determine the average torque. However, instead of computing a single overall average, which could introduce bias due to the presence of three different machines on each floor, a more accurate approach is to calculate the average torque for each machine separately. This ensures that we account for variations between different machines and provide a more precise representation of the data.

Torque is a crucial measurement in industrial settings, representing the rotational force applied to a machine. It directly impacts the efficiency, performance, and maintenance needs of machinery. Understanding the average torque for each machine allows us to identify potential operational inconsistencies and optimize machine performance.

To enhance interpretability, we will visualize the average torque per machine using a well-structured bar chart. This visualization will include appropriate color schemes, annotations, and a clear layout to make the insights more accessible and actionable.

1.2.1 How close are the troque values to the average?

While the average torque gives us an overall idea of the force exerted by each machine, it doesn't tell us how consistent the readings are over time. This is where standard deviation comes in.

- A low standard deviation means the torque values are closely packed around the mean, indicating consistent machine performance.
- A high standard deviation means the torque values fluctuate more, suggesting variability in machine operation, which could be due to load changes, tool wear, or operational inconsistencies.

By analyzing standard deviation, we can determine whether a machine maintains stable performance or if there are significant variations that might require further investigation.

Torque is a measure of the rotational force applied to a machine, expressed in Newton-meters (Nm). It plays a crucial role in determining machine efficiency, stability, and overall performance. In our analysis, we computed the average torque along with the standard deviation for each machine unit:

- Makino-L1-Unit1-2013: 24.98 Nm ± 6.07 Nm
- Makino-L2-Unit1-2015: 25.21 Nm ± 6.22 Nm
- Makino-L3-Unit1-2015: 25.56 Nm ± 6.07 Nm

The average torque tells us the typical force exerted by each machine, while the standard deviation (SD) provides insight into how much the torque values fluctuate over time.

- A higher standard deviation (like 6.22 Nm for Makino-L2) indicates greater variation in torque readings, which could be due to changing loads, machine wear, or inconsistent operation.
- A lower standard deviation suggests more stable and predictable performance.

By analyzing both metrics together, we can assess machine consistency, detect potential inefficiencies, and ensure optimal operation.

Makino-L2-Unit1-2015 has the highest standard deviation (6.22 Nm), meaning its torque values fluctuate the most. **Makino-L1 and Makino-L3** have a standard deviation of around 6.07 Nm, slightly lower than Makino-L2.

Possible Causes:

- Inconsistent workload distribution
- Tool wear or improper calibration
- Irregular material properties affecting

Type *Markdown* and LaTeX: α^2

In [6]:

```
# get the average torque
# Grouping by machine and calculating average torque
avg_torque_per_machine = machine_ori.groupby('Machine_ID')['Torque(Nm)'].a
avg_torque_per_machine
```

Out[6]:

	Machine_ID	mean	std
0	Makino-L1-Unit1-2013	24.989764	6.075603
1	Makino-L2-Unit1-2015	25.210580	6.219515
2	Makino-L3-Unit1-2015	25.567039	6.078901

1.3 Assembly Line No with highest reading of machine failures

In addition to analyzing torque, we also examined the number of machine failures across different assembly lines. Machine failures can be caused by various factors, such as mechanical wear, excessive loads, improper calibration, or environmental conditions. Below are the recorded failures for each assembly line:

- Shopfloor-L1: 454 failures (Highest)
- Shopfloor-L3: 415 failures
- Shopfloor-L2: 396 failures (Lowest)

Key Observations:

- Shopfloor-L1 has the highest number of failures (454), indicating that machines in this section may be experiencing higher stress, improper maintenance, or operational inefficiencies.
- Shopfloor-L3 is slightly better but still has a significant failure count (415).
- Shopfloor-L2 has the lowest failure count (396), suggesting it may have better maintenance or less intensive workloads.

In [7]:

```
# get Number of machine failure that occur on each assembly line
machine_failure_reading = machine_ori[machine_ori['Downtime'] =='Machine_F
                          groupby('Assembly_Line_No')['Downtime'].value_co

# sort the number of machine failure for each assembly line in decending o
machine_failure_reading.sort_values(by = 'count', ascending=False)
```

Out[7]:

	Assembly_Line_No	Downtime	count
0	Shopfloor-L1	Machine_Failure	450
2	Shopfloor-L3	Machine_Failure	411
1	Shopfloor-L2	Machine_Failure	396

Correlation Analysis

Key Observations

1️⃣ Strongest Positive Correlations

- Torque (Nm) & Hydraulic Pressure (Pa) → (0.17)
- This suggests that as hydraulic pressure increases, torque also tends to increase.
- Cutting Force (N) & Spindle Speed (Rps) → (0.24)
- Higher spindle speeds are associated with an increase in cutting force.
- Coolant Pressure (Pa) & Cutting Force (N) → (0.18)

A slight positive correlation suggests that higher coolant pressure may correspond with increased cutting force.

2️⃣ Notable Negative Correlations

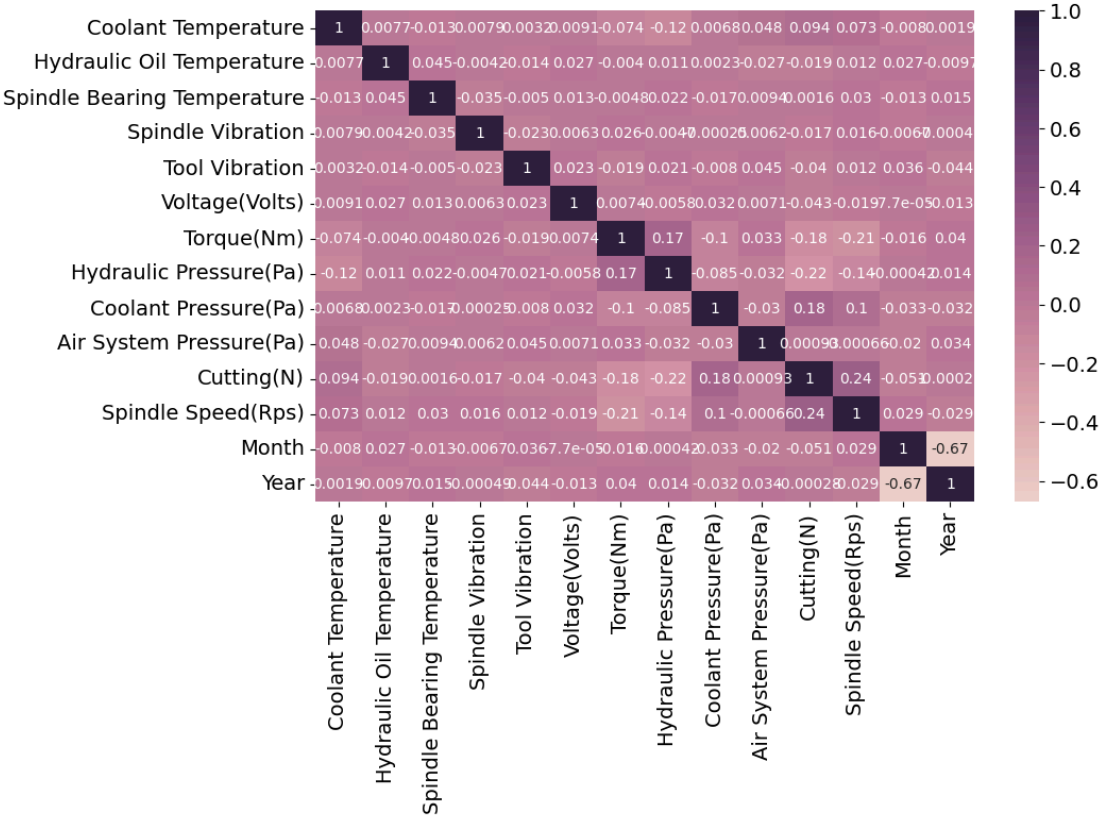
- Torque (Nm) & Cutting Force (N) → (-0.18)
- This could indicate that when torque increases, cutting force tends to reduce slightly, possibly due to tool efficiency or load balancing mechanisms.
- Spindle Speed (Rps) & Torque (Nm) → (-0.21)

This suggests that as spindle speed increases, the torque applied to the machine decreases, which is expected in high-speed machining operations. Cutting Force (N) & Hydraulic Pressure (Pa) → (-0.22) This might indicate that when hydraulic pressure is higher, the cutting force needed is lower, possibly due to improved lubrication or stabilization effects.

3️⃣ Weak/No Correlation Most other variables have very weak correlation values (close to 0), indicating little to no linear relationship. For instance:

- Coolant Temperature vs. Spindle Bearing Temperature (-0.013) suggests minimal dependency.
- Voltage vs. Spindle Vibration (0.0063) shows no meaningful relationship.

```
In [8]:
# copy the data
machine_ori_encoded = machine_ori.copy()

# create a custom
h_labels = [x.replace('_', ' ').title() for x in
            list(machine_ori_encoded.select_dtypes(include=['number']).col

# Select only numeric columns
numeric_df = machine_ori_encoded.select_dtypes(include=['number'])

fig, ax = plt.subplots(figsize=(10,6))
_ = sns.heatmap(numeric_df.corr(), annot=True,
                xticklabels=h_labels, yticklabels=h_labels,
                cmap=sns.cubehelix_palette(as_cmap=True), ax=ax)
```



Reliability Analysis of Each Machine

To evaluate the reliability of each machine, you can analyze Mean Time Between Failures (MTBF) and Failure Rate.

1. Calculate Time Between Failures (TBF)**

> Group data by Machine_ID.

> Find the time difference between successive failures.

The NaN values in the Time_Diff column suggest that:

> NaN values: These appear for the first failure entry of each machine because there’s no previous failure to calculate a time difference. Since that is the case we could replace the NaN with 0 to help our next calculation. We woul do that when calculating mean time betwen failures

In [9]:

```
machine_failure = machine_ori[machine_ori['Downtime'] == 'Machine_Failure
# Drop duplicate failures on the same date for the same machine
machine_failure = machine_failure.drop_duplicates(subset=['Machine_ID', 'D

# Compute Time Between Failures (TBF)
machine_failure['Time_Diff'] = machine_failure.groupby('Machine_ID')['Date

print(machine_failure[['Machine_ID', 'Date', 'Time_Diff']].head(20))
```

```
             Machine_ID       Date  Time_Diff
2   Makino-L1-Unit1-2013 2021-12-17        NaN
4   Makino-L2-Unit1-2015 2021-12-21        NaN
10  Makino-L1-Unit1-2013 2021-12-27       10.0
11  Makino-L3-Unit1-2015 2021-12-27        NaN
14  Makino-L2-Unit1-2015 2021-12-28        7.0
16  Makino-L1-Unit1-2013 2021-12-28        1.0
19  Makino-L1-Unit1-2013 2021-12-30        2.0
21  Makino-L1-Unit1-2013 2021-12-31        1.0
23  Makino-L3-Unit1-2015 2021-12-31        4.0
24  Makino-L1-Unit1-2013 2022-01-03        3.0
27  Makino-L3-Unit1-2015 2022-01-06        6.0
28  Makino-L2-Unit1-2015 2022-01-06        9.0
34  Makino-L1-Unit1-2013 2022-01-07        4.0
35  Makino-L2-Unit1-2015 2022-01-08        2.0
38  Makino-L1-Unit1-2013 2022-01-08        1.0
40  Makino-L3-Unit1-2015 2022-01-09        3.0
41  Makino-L2-Unit1-2015 2022-01-09        1.0
44  Makino-L1-Unit1-2013 2022-01-10        2.0
48  Makino-L1-Unit1-2013 2022-01-11        1.0
49  Makino-L3-Unit1-2015 2022-01-11        2.0
```

2. Compute MTBF (Mean time between Failures) for Each Machine

Mean time between failure (MTBF) is a measure of the reliability of a system or component. It’s a crucial element of maintenance management, representing the average time that a system or component will operate before it fails.

The MTBF formula is often used in the context of industrial or electronic system maintainability, where failure of a component can lead to significant downtime or even safety risks, but MTBF is used across many types of repairable systems and diverse industries.

It can help measure the overall reliability of manufacturing plants, energy grids, information networks and countless other use cases. ('https://www.ibm.com/think/topics/mtbf')['IBM'](https://www.ibm.com/think/topics/mtbf')%5B'IBM'%5D)

Key Insights & Recommendations:

Makino-L1-Unit1-2013 has the highest MTBF (1.584 days)

- This machine runs the longest before breaking down.
- Recommendation: Keep maintenance schedules the same or optimize them further.

Makino-L3-Unit1-2015 has the lowest MTBF (1.366 days)

- This machine fails the most frequently.
- Recommendation: Investigate possible causes (wear and tear, overheating, vibration, or improper usage). Preventive maintenance should be increased.

MTBF Differences are Small (~0.2 days apart)

- The variation in MTBF is not very large, but Makino-L3 needs attention.
- You may analyze failure patterns further: Does it fail under specific conditions (high spindle speed, high cutting force, etc.)?

In [10]:

```
# replace NaN with zero
machine_failure['Time_Diff'] = machine_failure['Time_Diff'].replace('NaN',
# compute MTBF
mtbf = machine_failure.groupby('Machine_ID')['Time_Diff'].mean()
print(mtbf)
```

```
Machine_ID
Makino-L1-Unit1-2013    1.584000
Makino-L2-Unit1-2015    1.539823
Makino-L3-Unit1-2015    1.366071
Name: Time_Diff, dtype: float64
```

In [11]:

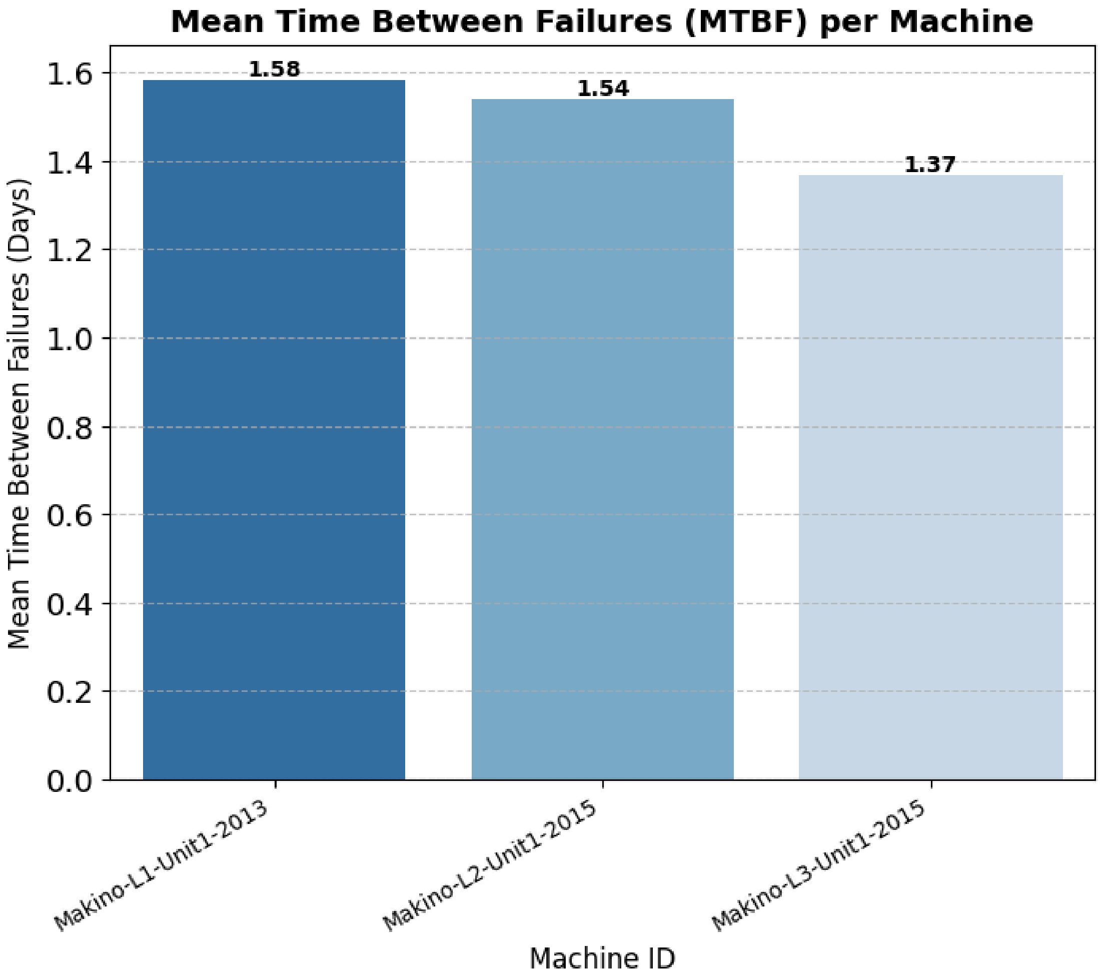
```
mtbf.head()
```

Out[11]:

```
Machine_ID
Makino-L1-Unit1-2013    1.584000
Makino-L2-Unit1-2015    1.539823
Makino-L3-Unit1-2015    1.366071
Name: Time_Diff, dtype: float64
```

In [12]:

```
# sort the MTBF in descending order
mtbf = pd.DataFrame(mtbf)
mtbf = mtbf.sort_values(by = 'Time_Diff', ascending= False)

# Create a Beautiful Plot
plt.figure(figsize=(8, 6))
cmap = sns.color_palette("coolwarm", as_cmap=True)  # Gradient colors
sns.barplot(data=mtbf, x='Machine_ID', y='Time_Diff', palette="Blues_r")

# Title and labels
plt.title('Mean Time Between Failures (MTBF) per Machine', fontsize=14, fo
plt.xlabel('Machine ID', fontsize=12)
plt.ylabel('Mean Time Between Failures (Days)', fontsize=12)
plt.xticks(rotation=30, ha="right", fontsize=10)

# Add value labels
for index, row in enumerate(mtbf['Time_Diff']):
    plt.text(index, row + 0.01, f'{row:.2f}', ha='center', fontsize=10, fo

# Show the plot
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```



3. Compute Failure Rate

What is Failure Rate? Failure rate (λ) measures how frequently a machine fails per unit time (e.g., failures per day). A higher failure rate means the machine fails more often and is less reliable. It is calculated as follows

Failure Rate (λ) = Total Failures / Total Operating Time

Key Insights & Recommendations:

1. Makino-L1-Unit1-2013 has the highest failure rate (0.6087 failures per day)

- This machine is the least reliable.
- Recommendation: Increase preventive maintenance and inspect for common failure causes (e.g., overheating, vibration, pressure issues).

2. Makino-L3-Unit1-2015 has the lowest failure rate (0.5459 failures per day)2

- This machine is the most reliable.
- Recommendation: Keep maintenance strategies consistent to maintain reliability.

> Failure rates are close in value (~0.06 difference)

The variation in failure rate is small, but Makino-L1 is still the most problematic. Investigate if failures are related to specific operational conditions (e.g., high spindle speed, torque, or

In [13]:

```python
# claculate the total operating time
total_time = (machine_ori['Date'].max() - machine_ori['Date'].min()).days
failure_counts = machine_failure['Machine_ID'].value_counts() # number of

failure_rate = failure_counts / total_time # claculat failure rate
print(failure_rate)    # print the failure rate
```

```
Machine_ID
Makino-L1-Unit1-2013    0.608696
Makino-L2-Unit1-2015    0.550725
Makino-L3-Unit1-2015    0.545894
Name: count, dtype: float64
```

Weibull Analysis for Machine Reliability

Weibull analysis helps us model failure behavior and determine whether failures are due to early-life issues, random occurrences, or wear-out over time. It uses the Weibull distribution, which is defined by two key parameters:

- Shape Parameter (β) → Determines the failure pattern
- Scale Parameter (η) → Represents the characteristic life (time when ~63% of units fail

In other words, for η, the higher this parameter the faster a failure occurs. Beta instead is a parameter that determines the shape of our distribution. Beta <1 : we have less and less failures with time as the weak part of our population is weeded out.

- β = 1 constant failure rate.
- β > 1: the failure rate increases with time as the population ages significantly.

The weibull density distribution function can be represented as follows:

$$F(x) = \frac{\beta}{\eta}\left(\frac{x}{\eta}\right)^{\beta-1} e^{-(x/\eta)^{\beta}}$$

1: Fit Weibull Distribution

We'll estimate the shape (β) and scale (η) parameters for each machine. We will be using the machine failure data for this as it works with the time to failure values

This plot represents the Weibull reliability analysis of the machine failure data. Let's break it down:

1. Axes Explanation

> - X-axis (Time to Failure, log scale): This represents the time intervals between failures, plotted on a logarithmic scale.
> - Y-axis (Unreliability %): This represents the probability of failure, showing how likely it is that a machine has failed by a given time.

2. Weibull Parameters (MLE Estimates)

> - β (Shape Parameter): 1.219 (MLE C4), 1.224 (Uncorrected MLE)
> - Since β < 1.5, this suggests an early-life failure trend (infant mortality). Failures are more likely due to defects, improper setup, or wear-in effects.

3. η (Scale Parameter): 1.686

> The scale parameter represents the characteristic life. About 63.2% of failures occur before this time.

4. Key Observations

- Straight-Line Fit: The points closely follow a straight line, confirming that the Weibull distribution is a good model for the data.
- Fisher Bounds (Confidence Intervals): The light blue lines represent a 90% confidence interval, indicating the expected variability in failure rates.
- Cluster of Points on the Right: The dense cluster at higher unreliability suggests that a significant number of failures happen in a short period.

5. Inference and Recommendations

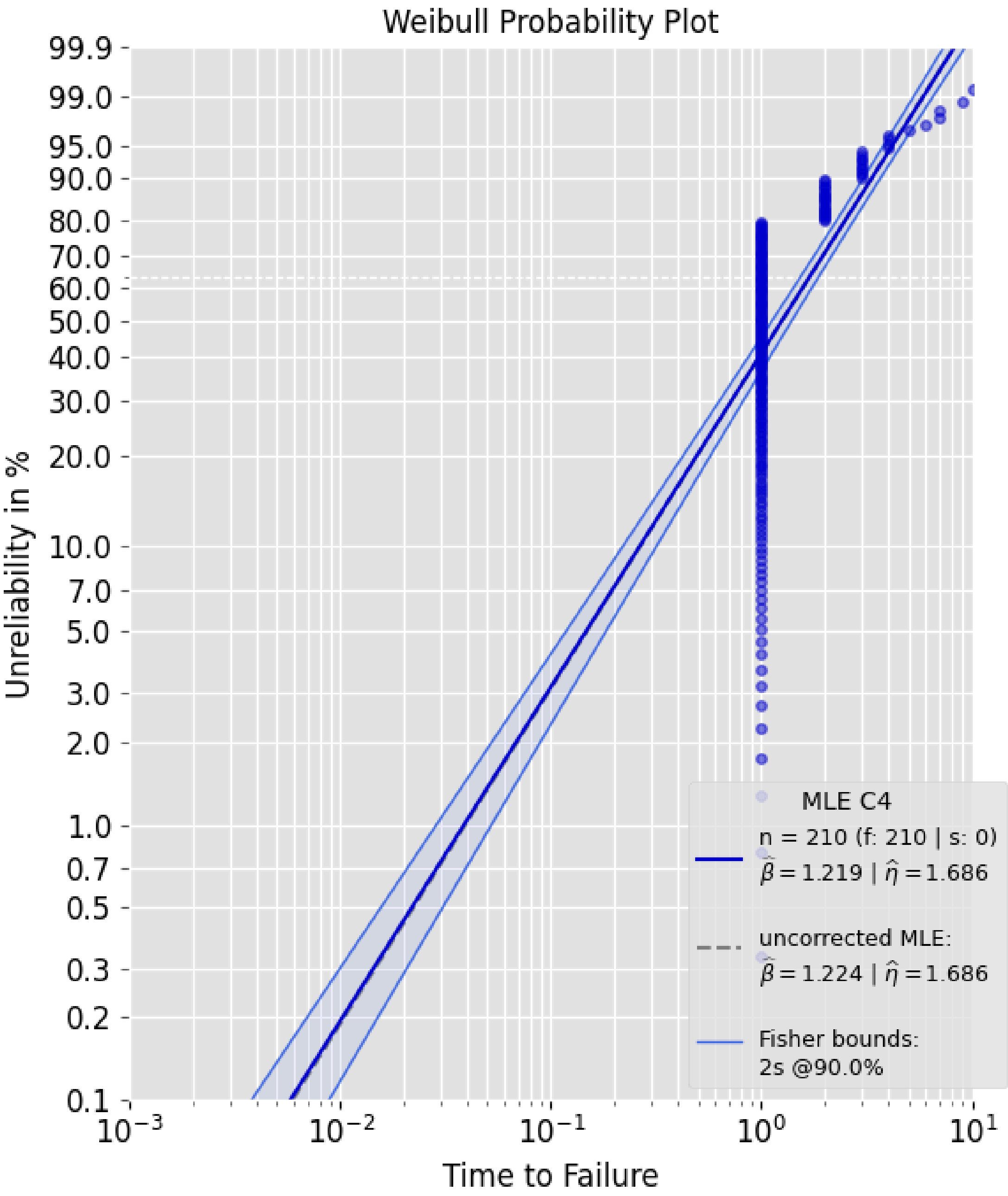
- Early Failures: The β value (~1.143) indicates that failures are due to infant mortality rather than wear-out. Preventive maintenance strategies should focus on initial quality control and better installation procedures.
- Short Characteristic Life (η ~1.732): Failures happen relatively quickly, so frequent monitoring and early intervention are necessary to improve uptime.
- Potential Design or Operational Flaws: Since failures happen sooner than expected, the machines might have design weaknesses, improper usage, or lack of preventive

maintenance measures.

- Reliability Growth Program Needed: To improve reliability, consider burn-in testing, redesigning weak components, or optimizing operational conditions.

In [14]:

```
machine_failure = machine_failure[machine_failure['Time_Diff'] > 0 ]
machine_failure = machine_failure.groupby('Machine_ID', group_keys=False)\
                  .apply(lambda x: x.sample(frac = 0.6))

prototype_a = Analysis(df=machine_failure['Time_Diff'], bounds='fb', show=
prototype_a.beta_init = 1.5  # Set an initial guess
prototype_a.mle()
```



The previous plot shows the overall reliability. we need to drill down to each machine characteristics by plotting separate weibull plot for them and then make a comparison

Time Series Analysis

✅ The key features to focus on:

- Torque (Nm), Hydraulic Pressure (Pa), Cutting Force (N), Spindle Speed (Rps), Coolant Pressure (Pa)
- Machine Failure occurrences (target variable)

1. Average Weekly values

This visualization presents weekly trends of five key machine parameters over time, including their average values and variability. Each subplot corresponds to a different feature, with the solid red line representing the average value, while the shaded regions indicate the range (min-max values) for each week. Here’s a breakdown:

1. Hydraulic Pressure (Pa)

- Steady upward trend in pressure over time, peaking before declining slightly.
- High variability in the early weeks, then a more consistent range.

2. Cutting Force (N)

- Shows a sharp increase initially, stabilizing afterward.
- Variability decreases, indicating more controlled cutting conditions.

3. Spindle Speed (RPS)

- Sudden drop in the first few weeks, then stabilizes with slight fluctuations.
- The min-max range narrows over time, suggesting more uniform operations.

4. Coolant Pressure (Pa)

- High fluctuations initially, stabilizing before a later drop.
- Suggests initial tuning of the coolant system before achieving consistency.

5.Torque (Nm)

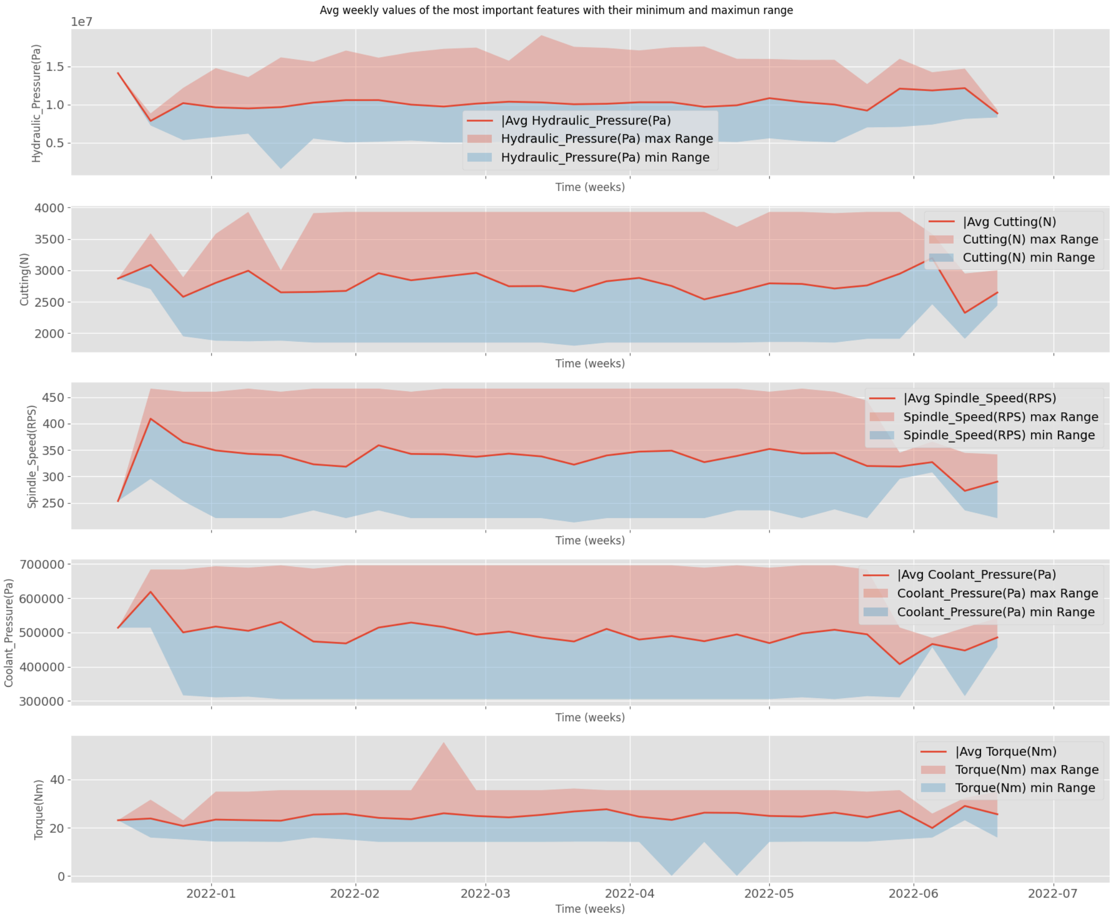
- Relatively stable trend with slight fluctuations.
- A spike in March suggests a period of increased torque demand.

Key Insights:

- Early periods show high variability across all parameters, indicating initial adjustments or unstable conditions.
- Most parameters stabilize over time, suggesting improved control and machine performance.
- Spindle speed and cutting force appear correlated, possibly indicating a process dependency.

In [15]:

```
# Ensure the timestamp column is in datetime format (Modify 'Timestamp' to
machine_ori['Date'] = pd.to_datetime(machine_ori['Date'])

# Set timestamp as index
machine_ori.set_index('Date', inplace=True)
# Resampling data (weekly mean)
sample_window = '1W'
# Select relevant numeric columns
features = ['Hydraulic_Pressure(Pa)', 'Cutting(N)', 'Spindle_Speed(RPS)',
            'Coolant_Pressure(Pa)', 'Torque(Nm)']
machine_ori[features] = machine_ori[features].apply(pd.to_numeric, errors=

# drop the NAN Values
machine_ori = machine_ori.dropna(subset=features)

# resampling to weely statistics
weekly_stats = machine_ori[features].resample(sample_window).agg(['mean',
                                                                   'max',

# create sybplot (one for time series, one for bar chart)
#num_features = len(features)
fig, axes = plt.subplots(len(features), 1, figsize=(18, 15),
                         sharex=True)

# Ensure axes is always iterable
#if num_features == 1:
#    axes = [axes]

# plot each feature time series with shaded region
for i, feature in enumerate(features):
    ax = axes[i]
    ax.plot(weekly_stats.index, weekly_stats[(feature, 'mean')],
            lw = 2, label = f'|Avg {feature}')
    ax.fill_between(weekly_stats.index, weekly_stats[(feature, 'max')],
                    weekly_stats[(feature, 'mean')], alpha = 0.3,
                    label = f'{feature} max Range')
    ax.fill_between(weekly_stats.index, weekly_stats[(feature, 'min')],
                    weekly_stats[(feature, 'mean')], alpha = 0.3,
                    label = f'{feature} min Range')

    # formatting the time series plot
    ax.set_ylabel(f'{feature}')
    ax.set_xlabel('Time (weeks)')
    ax.legend()
    ax.grid(True)
    fig.suptitle('Avg weekly values of the most important features with th
    plt.tight_layout()
```



2. Monthly Variation of Features based on downtime

This bar chart presents the monthly variation of key machine performance features while distinguishing between Machine Failure and No Machine Failure.

Key Observations by Feature: **1. Hydraulic Pressure (Pa)**

- Generally higher when the machine is running without failure.
- During failure, the pressure tends to drop, especially in December and June, suggesting that low hydraulic pressure may be linked to machine failures.

2. Cutting Force (N)

- The force is higher in months without failure.
- Lower cutting force during failure events suggests that either the machine struggles to maintain force or failure occurs at lower operational levels.

3. Spindle Speed (RPS)

- Spindle speed is significantly higher during machine failures, particularly in December.
- This suggests that excessive spindle speeds may contribute to machine breakdowns, possibly due to overheating or mechanical stress.

4. Coolant Pressure (Pa)

- Higher coolant pressure is observed in failure conditions, particularly in December.

- This could indicate excessive cooling system activation when failures are about to occur, possibly due to overheating compensation.

5. Torque (Nm)

- Torque is generally lower during machine failure months.
- A noticeable increase in torque is seen in June under failure conditions, suggesting that excessive torque might be a failure precursor in certain months.

General Insights:

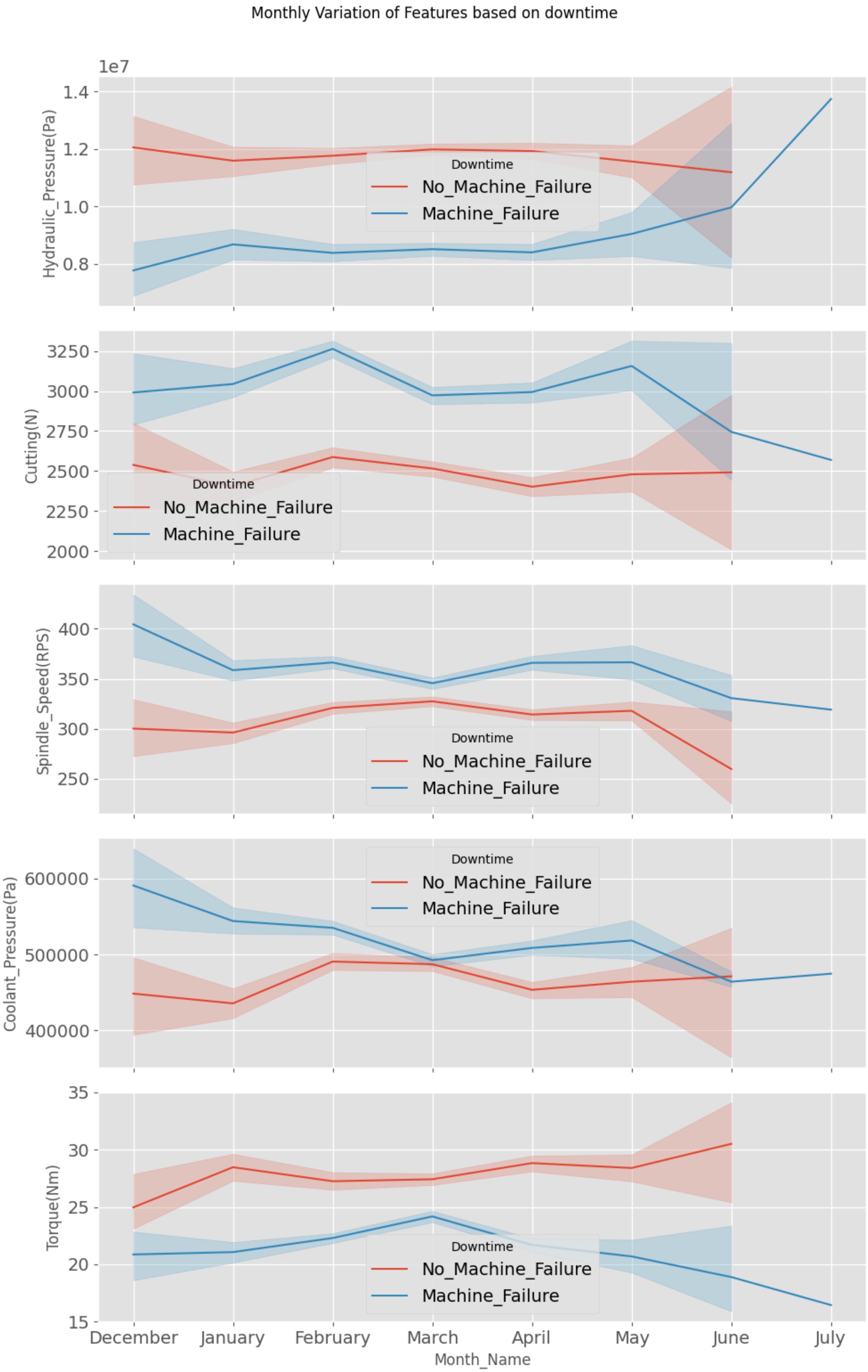
> Hydraulic Pressure & Cutting Force Drop Before Failure: Suggests lubrication or material engagement issues.

> High Spindle Speed & Coolant Pressure During Failures: Could indicate stress-related overheating or instability.

> Torque Variations Indicating Failure Patterns: High torque could be a failure trigger in some cases.

In [16]:

```
# Create seaprate bar plot for each feature
fig, axes = plt.subplots(len(features), 1,
                         figsize = (10, 15), sharex=True)

for i, feature in zip(axes, features):
    sns.lineplot(data = machine_ori, x = 'Month_Name',hue = 'Downtime',
                 y = feature, estimator='mean', ax = i)

    ax.set_ylabel(f'{feature}')
    ax.set_xlabel('Month')
    ax.set_title('Monthly variation of {feature}')

plt.tight_layout()
plt.suptitle('Monthly Variation of Features based on downtime', y = 1.03)
plt.show();
```



1. Variation of Hydraulic Pressure per month based on machine type and whether there is Downtime

Specific Machine Insights:

1. Makino-L2-Unit1-2015:

- High Pressure in January (No Failure): There's a noticeable spike in hydraulic pressure during January when there was no machine failure. This could indicate a specific operational pattern or external factor affecting the machine during that month.
- Fluctuations During Failure: During downtime periods, there seems to be more variability in hydraulic pressure, as indicated by the larger error bars. This might be expected as the system is not operating under normal controlled conditions.

Makino-L1-Urvt1-2013:

- Lower Overall Pressure: The overall hydraulic pressure for this machine appears to be lower compared to the other two machines. This could be due to design differences, operating conditions, or the specific type of component it produces.
- Pressure Drop During Failure (July): In July, there's a significant drop in pressure during a downtime period. This suggests a potential link between low hydraulic pressure and the reason for the machine's failure.

Makino-L3-Unit1-2015:

- Stable Pressure (No Failure): The hydraulic pressure during non-failure periods seems relatively stable across the months for this machine.
- Potential Pressure Increase During Failure (July): In July, there's a noticeable increase in pressure during a downtime period. This is different from the "Makino-L1" machine and suggests that different types of failures might manifest in different hydraulic pressure patterns.

2. Variation of Cutting Force per month based on machine type and whether there is Downtime

The second visual displays a series of bar graphs showing the variation of cutting force (measured in Newtons, N) over several months for three different machines: "Makino-L2-Unit1-2015", "Makino-L1-Unit1-2013", and "Makino-L3-Unit1-2015". Each graph compares the cutting force during periods of "No_Machine_Failure" with periods of "Machine_Failure" (downtime).

Here's an interpretation of the key observations and insights:

General Observations:

- Cutting Force Trends: The graphs illustrate how the cutting force fluctuates month-to-month for each machine and how these fluctuations relate to downtime.
- Direct Comparison: The side-by-side arrangement of the bars makes it easy to visually compare the cutting force during operational periods versus downtime periods within each month.
- Error Bars as Variability: The error bars associated with each bar represent the variability (likely standard deviation or confidence interval) of the cutting force measurements within that specific month and downtime category. Larger error bars indicate more fluctuation or less consistent readings.

Machine Specific Insights:

1. Makino-L2-Unit1-2015:

- Relatively Stable Operation: During periods of no machine failure, the cutting force appears relatively stable across the months, with some minor variations.
- Potential Decrease During Downtime: In some months (e.g., February, April), there appears to be a decrease in the average cutting force during downtime periods compared to operational periods. However, the error bars suggest that this difference might not be statistically significant in all cases.

2. Makino-L1-Unit1-2013:

- More Fluctuations: This machine seems to exhibit more variability in cutting force, even during operational periods, compared to the other two machines.
- No Clear Downtime Pattern: There isn't a consistent pattern of increase or decrease in cutting force during downtime periods for this machine. The changes are less pronounced and less consistent than in the other machines.

3. Makino-L3-Unit1-2015:

- Higher Cutting Force: This machine appears to operate at a higher average cutting force compared to the other two machines.
- Potential Decrease During Downtime: Similar to the "Makino-L2" machine, there's a suggestion of a decrease in cutting force during some downtime periods (e.g., June), but again, the error bars indicate the need for caution in interpreting these differences.

3. Variation of Spindle Soeed per month based on machine type and whether there is Downtime

Machine-Specific Insights:

1. Makino-L2-Unit1-2015:

- Potential Decrease in Speed During Downtime: In several months (e.g., January, February, April, June), there's a visual suggestion of a decrease in average spindle speed during downtime periods compared to operational periods.
- Variability in Operational Speed: The spindle speed during "No_Machine_Failure" periods shows some degree of variation across the months, which could be due to different operating conditions or tasks.

2. Makino-L1-Unit1-2013:

- Less Consistent Pattern: The relationship between spindle speed and downtime is less clear for this machine. In some months, the speed appears lower during downtime, while in others, it's relatively similar or even slightly higher.
- Significant Speed Drop in July (Downtime): A notable decrease in spindle speed is observed in July during a downtime period. This could indicate a specific issue affecting the spindle or related systems.

3. Makino-L3-Unit1-2015:

- Similar Trends to Makino-L2: This machine shows some similarities to the "Makino-L2" machine, with a potential tendency towards lower spindle speeds during downtime periods in some months.
- Speed Variation in June (Both Categories): There's a significant variation in spindle speed in June, both during operational and downtime periods, suggesting potential instability or changes in operating parameters.

4. Variation of Spindle Speed per month based on machine type and whether there is Downtime

Machine-Specific Insights:

1. Makino-L2-Unit1-2015:

- Relatively Stable Pressure: During periods of no machine failure, the coolant pressure appears relatively stable across the months, with some minor variations.
- Potential Pressure Decrease During Downtime: There's a suggestion of a decrease in average coolant pressure during downtime periods in some months (e.g., January, February, April, June). However, the error bars indicate that these differences might not be statistically significant.

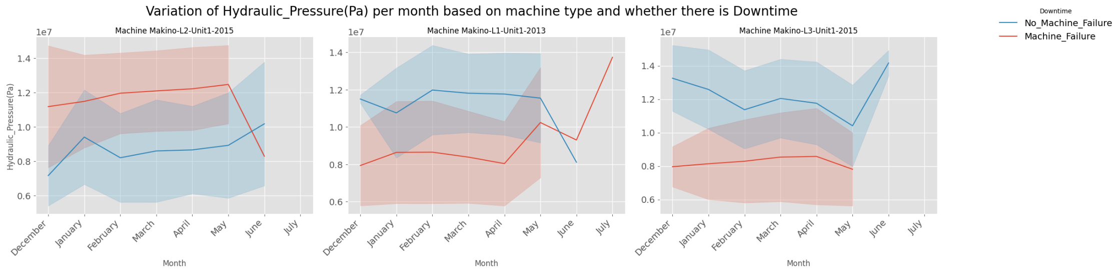
2. Makino-L1-Unit1-2013:

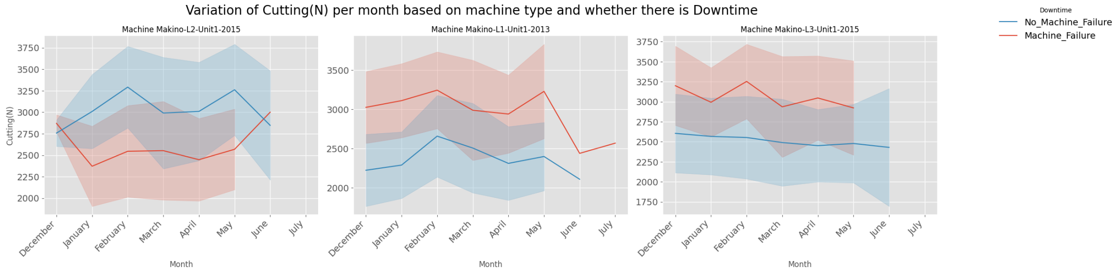
- More Fluctuations: This machine exhibits more variability in coolant pressure, even during operational periods, compared to the other two machines.
- No Clear Downtime Pattern: There isn't a strong or consistent pattern of increase or decrease in coolant pressure during downtime periods. The changes are less pronounced and less consistent than in the other machines.
- Potential Drop in July (Downtime): There's a possible decrease in coolant pressure during downtime in July, but it's important to consider the error bars and potential statistical significance.

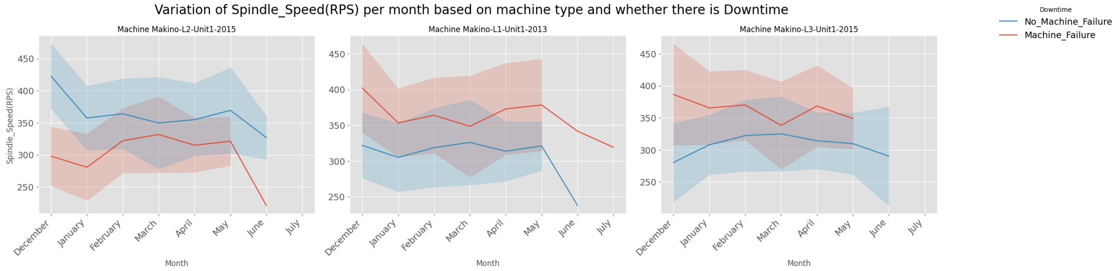
3. Makino-L3-Unit1-2015:

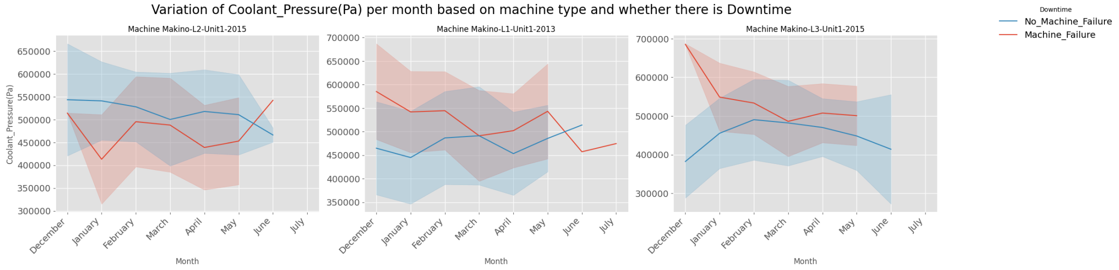
- Potential Pressure Increase During Downtime: In some months (e.g., February), there's a suggestion of an increase in coolant pressure during downtime. This is different from the "Makino-L2" machine and suggests that different types of failures might manifest in different coolant pressure patterns.
- High Pressure in December (No Failure): This machine shows a high coolant pressure during operational periods in December. This warrants further investigation to understand if it's a normal operating condition or a potential anomaly.

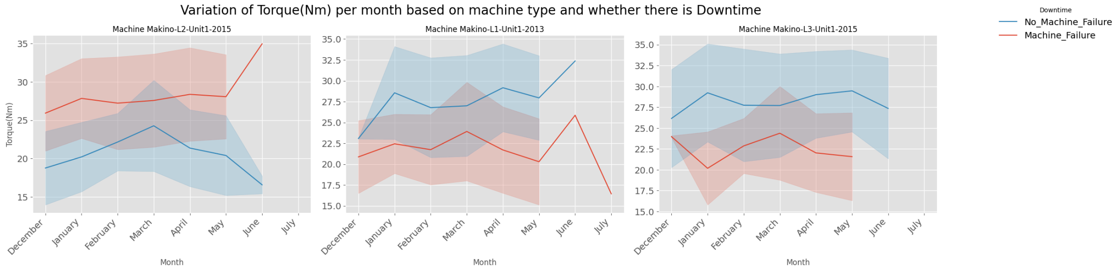
```
In [18]:  # Loop through each feature to create a FacetGrid
          for feature in features:
              g = sns.FacetGrid(machine_ori, col="Machine_ID", col_wrap=3,
                                height=6, sharey=False)  # Create separate plots for
              g.map_dataframe(sns.lineplot, x="Month_Name", y=feature, hue="Downtime

              for ax in g.axes.flat:
                  ax.set_xticklabels(ax.get_xticklabels(), rotation=45, ha="right")

              # Move legend outside the grid to avoid obstruction
              g.add_legend(title="Downtime", bbox_to_anchor=(1.05, 1), loc="upper le

              # Adjust plot properties
              g.set_axis_labels("Month", feature)
              g.set_titles(col_template="Machine {col_name}")

              plt.xticks(rotation=45)
              plt.suptitle(f'Variation of {feature} per month based on machine type
              plt.tight_layout()
              plt.show()
```


Variation of Hydraulic_Pressure(Pa) per month based on machine type and whether there is Downtime


Variation of Cutting(N) per month based on machine type and whether there is Downtime


Variation of Spindle_Speed(RPS) per month based on machine type and whether there is Downtime


Variation of Coolant_Pressure(Pa) per month based on machine type and whether there is Downtime



Statistical Analysis

ANOVA or Kruskal-Wallis Test (comparing sensor values between failure cases)

- P-values greater than 0.05 (common significance level) indicate that we fail to reject the null hypothesis.
- This means there is no significant difference in the distributions of the sensor values across different machine groups.
- Since all the p-values are above 0.05, we conclude that variations in sensor readings (like pressure, temperature, or spindle speed) are not significantly different among the machines in relation to failures.

In [47]:

```
from scipy.stats import kruskal

# Compare pressure across machines
machine_groups = [machine_ori[machine_ori["Machine_ID"] == m][features] fo

# calulate the stat and p-value
stat, p = kruskal(*machine_groups)
print(f"Kruskal-Wallis Statistic: {stat}, P-value: {p}")
```

```
Kruskal-Wallis Statistic: [4.004093   0.59165683 0.54061884 0.49393769
3.29262592], P-value: [0.1350586  0.74391506 0.76314333 0.78116503 0.192
75931]
```

Pearson Correlation Test (For Linear Relationships)

This test helps us determine whether the correlations seen in the heatmap are statistically significant.

- Null Hypothesis (H_0): No correlation between the variables.
- Alternative Hypothesis (H_1): There is a correlation between the variables. If the p-value < 0.05, we reject H_0 and say that the correlation is significant.

Explanation of Result

- All correlations are statistically significant (p-value = 0.00000), meaning the relationships are not due to random chance.
- The negative correlation between torque and spindle speed could suggest a tradeoff in machine performance.
- Spindle Speed & Cutting Force have the strongest correlation (0.244), suggesting they might be closely linked in machine operation.

In [48]:

```
# Select numerical features
numerical_features = ["Torque(Nm)", "Hydraulic_Pressure(Pa)", "Coolant_Pre
                      "Spindle_Speed(RPS)", "Cutting(N)"]

# Create an empty DataFrame to store results
correlation_results = []

# Compute Pearson correlation and p-values
for i in range(len(numerical_features)):
    for j in range(i + 1, len(numerical_features)):
        feature1, feature2 = numerical_features[i], numerical_features[j]
        correlation, p_value = stats.pearsonr(machine_ori[feature1], machi

        # Append results as a dictionary
        correlation_results.append({
            "Feature 1": feature1,
            "Feature 2": feature2,
            "Pearson Correlation": round(correlation, 3),
            "p-value": round(p_value, 5),
            "Significance": "Significant" if p_value < 0.05 else "Not Sign
        })

# Convert results into a Pandas DataFrame
correlation_df = pd.DataFrame(correlation_results)
correlation_df
```

Out[48]:

	Feature 1	Feature 2	Pearson Correlation	p-value	Significance
0	Torque(Nm)	Hydraulic_Pressure(Pa)	0.166	0.00000	Significant
1	Torque(Nm)	Coolant_Pressure(Pa)	-0.100	0.00000	Significant
2	Torque(Nm)	Spindle_Speed(RPS)	-0.208	0.00000	Significant
3	Torque(Nm)	Cutting(N)	-0.181	0.00000	Significant
4	Hydraulic_Pressure(Pa)	Coolant_Pressure(Pa)	-0.085	0.00002	Significant
5	Hydraulic_Pressure(Pa)	Spindle_Speed(RPS)	-0.139	0.00000	Significant
6	Hydraulic_Pressure(Pa)	Cutting(N)	-0.224	0.00000	Significant
7	Coolant_Pressure(Pa)	Spindle_Speed(RPS)	0.104	0.00000	Significant
8	Coolant_Pressure(Pa)	Cutting(N)	0.176	0.00000	Significant
9	Spindle_Speed(RPS)	Cutting(N)	0.244	0.00000	Significant

Categorical Feature (Machine_ID) and Categorical Target (Downtime)

For Machine_ID, you can use the Chi-Square test because both variables are categorical.

SInce p-value is > 0.05, this means that there is no statistically significant relationship between Machine_ID and Downtime. In other words, the machine ID does not strongly influence whether a failure occurs based on the given data.

What Does This Mean?

- It could be that The distribution of failures (Downtime) is not significantly different across different machines.
- Failures may be more dependent on numerical features (e.g., Torque(Nm), Spindle Speed(RPS)) rather than the machine ID itself.

In [49]:

```
# Create a contingency table
contingency_table = pd.crosstab(machine_ori["Machine_ID"], machine_ori["Do

# Chi-Square test
chi2, p, dof, expected = chi2_contingency(contingency_table)

print(f"Chi-Square Statistic: {chi2:.3f}, p-value: {p:.5f}")
print(" Significant relationship!" if p < 0.05 else " No significant relat
```

```
Chi-Square Statistic: 1.075, p-value: 0.58433
 No significant relationship.
```

Conclusion: Key Findings and Next Steps

Through an extensive exploratory data analysis, we identified key features that significantly impact machine downtime prediction. Based on Pearson correlation and Chi-Square tests, the most influential factors include:

- Torque (Nm): Shows a significant inverse correlation with Spindle Speed (RPS) and Cutting Force (N), indicating its impact on machine failures.
- Spindle Speed (RPS): Positively correlated with Cutting Force (N), suggesting that variations in spindle speed affect the overall machine performance.
- Hydraulic Pressure (Pa): Exhibits a significant relationship with Torque and Cutting Force, which suggests that fluctuations in hydraulic pressure can contribute to machine failures.
- Coolant Pressure (Pa): Though its direct correlation with failures is weak, it plays a supporting role in machine operations, influencing Torque and Cutting Force.
- Categorical Features (Machine ID, Assembly Line): The Chi-Square test showed that Machine ID alone is not a significant predictor of failures, but analyzing failures per machine could help in detecting patterns.

Key Takeaways for Model Building:

✅ Focus on numerical features (Torque, Spindle Speed, Cutting Force, Hydraulic Pressure, and Coolant Pressure) as primary predictors.

✅ Perform feature scaling (e.g., Standardization) to ensure numerical features are on the same scale for optimal model performance.

In [ ]: