

```

import pandas as pd

# Load the dataset
# Use Load function before this if you are using colab

df = pd.read_csv('path to dataset')
# 1. Display the first 10 rows
first_10_rows = df.head(10)

# 2. Find unique values in 'Type' column and count machines of each type
unique_types = df['Type'].unique()
type_counts = df['Type'].value_counts()

# 3. Select all records where Tool wear is greater than 50
high_tool_wear = df[df['Tool wear [min]'] > 50]

# 4. Rename columns
df = df.rename(columns={
    'Torque [Nm]': 'Torque_Nm',
    'Air temperature [K]': 'AirTemp_K'
})

# 5. Check for missing values in each column
missing_values = df.isnull().sum()

# 6. Filter rows where Rotational speed > 1600 and Torque < 30
filtered_rows = df[(df['Rotational speed [rpm]'] > 1600) & (df['Torque_Nm'] < 30)]

# 7. Create a new column Temp_Diff = Process temperature - Air temperature
df['Temp_Diff'] = df['Process temperature [K]'] - df['AirTemp_K']

# 8. Group by Type and compute average Tool wear and Torque
grouped_stats = df.groupby('Type')[['Tool wear [min]', 'Torque_Nm']].mean()

# 9. Sort by Tool wear descending and get top 5 rows
top_tool_wear = df.sort_values(by='Tool wear [min]',
                               ascending=False).head(5)

# 10. Create Failure_Flag column: 1 if any failure indicator is 1, else 0
df['Failure_Flag'] = (
    df[['TWF', 'HDF', 'PWF', 'OSF', 'RNF']].sum(axis=1) > 0

```

```
).astype(int)

# 11. Compute total machine failures for each Product ID and show products
with more than 1 failure
product_failures = df.groupby('Product ID')['Machine failure'].sum()
products_multiple_failures = product_failures[product_failures > 1]

# 12. Group by Type and Machine failure, compute average Torque and Tool
wear
multi_group = df.groupby(['Type', 'Machine failure'])[['Torque_Nm', 'Tool
wear [min]']].mean()

# 13. Replace zero rotational speeds with median of that column
median_rpm = df['Rotational speed [rpm]'].median()
df.loc[df['Rotational speed [rpm]' == 0, 'Rotational speed [rpm]'] =
median_rpm

# 14. Machine type with highest average Temp_Diff
avg_temp_diff = df.groupby('Type')['Temp_Diff'].mean()
type_highest_temp_diff = avg_temp_diff.idxmax()

# 15. Filter top 10% Tool wear rows, then group by Type and calculate
average Failure_Flag
top_10_percent_wear = df['Tool wear [min]'].quantile(0.9)
top_wear_df = df[df['Tool wear [min]'] > top_10_percent_wear]
avg_failure_rate_by_type =
top_wear_df.groupby('Type')['Failure_Flag'].mean()
```