Task 27- Manufacturing Production Data

Description: Factories track production units, machine status, defects, timestamps, and shifts. Managers want to optimize production efficiency.

DATASET:

record_id	factory_id	factory_name	latitude	longitude	production_line	machine_id	machine_type	timestamp	shift	units_produced	machine_hours	defects	defect_level	operator_id	operator_notes	date
R00000	F003	South Plant	12.9627477	77.59047812	Line-A	F003_Line-A_CNC_5	CNC	02-02-2025 10:16	Evening	105	6.44	1	Low	OP015	calibration	02-02-2025
R00001	F003	South Plant	12.9713554	77.59815551	Line-A	F003_Line-A_Lathe_9	Lathe	12-01-2025 17:57	Evening	75	7.51	2	Low	OP038	vibration power	12-01-2025
R00002	F001	North Plant	28.7154023	77.10623119	Line-A	F001_Line-A_CNC_4	CNC	07-03-2025 08:57	Day	136	8.05	0	None	OP039	power	07-03-2025
R00003	F003	South Plant	12.9498167	77.58416104	Line-A	F003_Line-A_Lathe_9	Lathe	23-02-2025 19:33	Day	56	5.38	1	Low	OP015	normal operation	23-02-2025
R00004	F002	East Plant	19.0786349	72.89411771	Line-C	F002_Line-C_CNC_1	CNC	22-01-2025 04:52	Day	126	8.79	8	High	OP045	normal operation	22-01-2025
R00005	F002	East Plant	19.0692308	72.88381676	Line-B	F002_Line-B_CNC_4	CNC	14-02-2025 09:40	Evening	109	6.79	0	None	OP007	none	14-02-2025
R00006	F002	East Plant	19.0609185	72.88869647	Line-A	F002_Line-A_Press_6	Press	20-03-2025 19:10	Evening	108	8.54	1	Low	OP017	normal operation	20-03-2025
R00007	F001	North Plant	28.7077164	77.0960488	Line-C	F001_Line-C_Press_2	Press	18-02-2025 23:02	Day	91	7.69	6	High	OP006	normal operation	18-02-2025
R00008	F003	South Plant	12.9724705	77.59160993	Line-B	F003_Line-B_Robot_10	Robot	17-02-2025 08:01	Evening	174	10.39	1	Low	OP037	vibration	17-02-2025
R00009	F003	South Plant	12.9635151	77.58958243	Line-A	F003_Line-A_CNC_4	CNC	08-02-2025 03:31	Night	56	3.47	1	Low	OP006	power	08-02-2025
R00010	F001	North Plant	28.7015337	77.09882174	Line-B	F001_Line-B_CNC_6	CNC	22-01-2025 17:15	Night	102	7.49	4	Medium	OP024	normal operation	22-01-2025
R00011	F002	East Plant	19.0669301	72.86952646	Line-A	F002_Line-A_Lathe_2	Lathe	20-03-2025 23:13	Night	58	5.8	2	Low	OP041	vibration	20-03-2025
R00012	F003	South Plant	12.9608747	77.58467414	Line-C	F003_Line-C_CNC_8	CNC	18-02-2025 17:56	Evening	96	6.07	1	Low	OP018	power overheat vibration	18-02-2025
R00013	F003	South Plant	12.9830282	77.60211933	Line-C	F003_Line-C_CNC_6	CNC	09-01-2025 03:42	Evening	99	6.5	1	Low	OP015	normal operation	09-01-2025
R00014	F001	North Plant	28.7119967	77.10681489	Line-B	F001_Line-B_Press_2	Press	29-01-2025 15:11	Day	81	7.18	3	Medium	OP037	jam none lubrication	29-01-2025
R00015	F002	East Plant	19.0802801	72.85300301	Line-A	F002_Line-A_Lathe_7	Lathe	24-03-2025 07:43	Evening	58	5.53	1	Low	OP030	normal operation	24-03-2025
R00016	F001	North Plant	28.7008551	77.10044133	Line-B	F001_Line-B_CNC_9	CNC	11-03-2025 10:01	Evening	126	8.5	2	Low	OP017	none misalignment jam	11-03-2025
R00017	F003	South Plant	12.9647998	77.59692254	Line-B	F003_Line-B_Robot_7	Robot	16-02-2025 10:00	Day	183	9.41	4	Medium	OP015	normal operation	16-02-2025
R00018	F001	North Plant	28.711445	77.09295503	Line-C	F001_Line-C_Press_1	Press	16-01-2025 06:53	Day	85	6.89	3	Medium	OP010	normal operation	16-01-2025
R00019	F003	South Plant	12.9691461	77.58706264	Line-A	F003_Line-A_Robot_7	Robot	18-03-2025 13:28	Evening	121	6.53	2	Low	OP005	normal operation	18-03-2025
R00020	F002	East Plant	19.1033442	72.87863765	Line-B	F002_Line-B_Lathe_8	Lathe	09-03-2025 11:10	Night	47	5.14	2	Low	OP017	normal operation	09-03-2025
R00021	F003	South Plant	12.9693654	77.60174	Line-A	F003_Line-A_Lathe_2	Lathe	29-03-2025 09:26	Evening	71	6.56	6	High	OP035	normal operation	29-03-2025
R00022	F002	East Plant	19.0667395	72.87257035	Line-C	F002_Line-C_Press_5	Press	26-02-2025 09:53	Night	78	6.01	2	Low	OP011	normal operation	26-02-2025
R00023	F002	East Plant	19.0765821	72.8662703	Line-A	F002_Line-A_Milling_5	Milling	07-03-2025 13:50	Evening	82	6.85	3	Medium	OP049	normal operation	07-03-2025
R00024	F001	North Plant	28.7071445	77.10507207	Line-C	F001_Line-C_Milling_5	Milling	24-03-2025 07:57	Evening	92	7.7	3	Medium	OP033	normal operation	24-03-2025
R00025	F003	South Plant	12.9684473	77.60218969	Line-A	F003_Line-A_CNC_3	CNC	12-03-2025 02:23	Evening	125	8.62	4	Medium	OP050	normal operation	12-03-2025
R00026	F003	South Plant	12.9674547	77.59540199	Line-A	F003_Line-A_Lathe_8	Lathe	03-01-2025 13:07	Evening	67	6.68	3	Medium	OP008	normal operation	03-01-2025
R00027	F002	East Plant	19.0926377	72.87030764	Line-B	F002_Line-B_CNC_4	CNC	14-03-2025 11:06	Evening	101	6.38	3	Medium	OP006	misalignment	14-03-2025
R00028	F003	South Plant	12.9682575	77.59142153	Line-B	F003_Line-B_Robot_9	Robot	18-01-2025 00:12	Day	150	7.07	1	Low	OP009	none sensor jam	18-01-2025
R00029	F002	East Plant	19.0972216	72.88802465	Line-C	F002_Line-C_CNC_9	CNC	20-03-2025 07:17	Evening	75	4.37	4	Medium	OP028	normal operation	20-03-2025
R00030	F001	North Plant	28.6911506	77.10144881	Line-C	F001_Line-C_Robot_4	Robot	09-02-2025 11:19	Day	150	7.38	1	Low	OP026	normal operation	09-02-2025
R00031	F003	South Plant	12.9832316	77.59470233	Line-C	F003_Line-C_Press_9	Press	27-02-2025 19:14	Evening	63	4.82	1	Low	OP008	normal operation	27-02-2025
R00032	F001	North Plant	28.6851262	77.11530541	Line-A	F001_Line-A_CNC_1	CNC	17-03-2025 09:38	Day	129	8.5	3	Medium	OP036	normal operation	17-03-2025

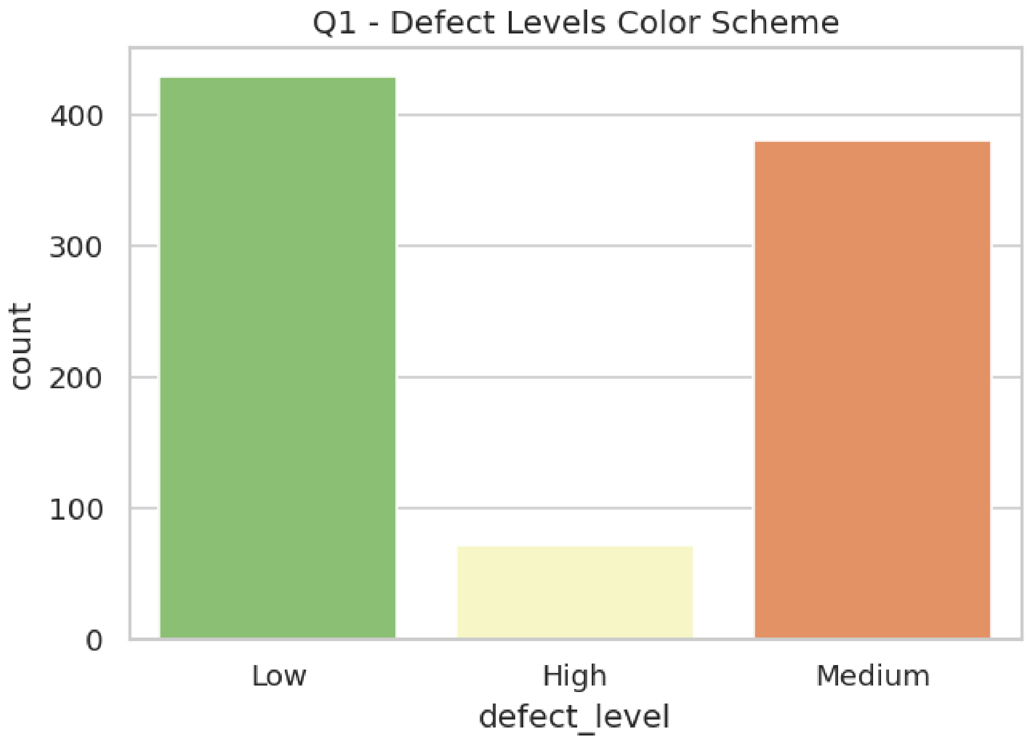
QUESTIONS:

1. Explain color schemes for defect levels.

Code:

```
plt.figure(figsize=(6,4))

sns.countplot(x='defect_level', data=df, palette='RdYlGn_r')

plt.title("Q1 - Defect Levels Color Scheme")

plt.show()
```

Visualization:



Inferences (Q1):

1. Red zones indicate critical defect levels (High) demanding attention.

2. Green zones show fewer minor or no defects.

3. Balanced color helps identify severity visually.

4. Helps managers quickly distinguish problem areas.

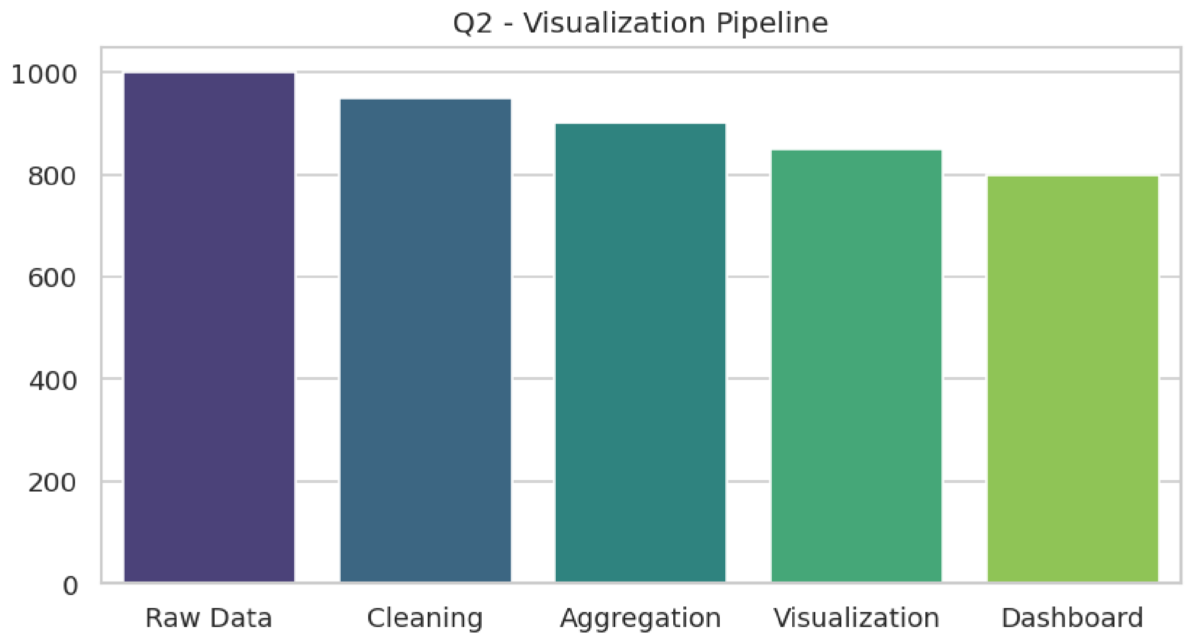
5. Shows distribution of defect severity across dataset.

2. Visualization pipeline from raw data to dashboards.

Code:

```
stages = ["Raw Data", "Cleaning", "Aggregation", "Visualization", "Dashboard"]

counts = [1000, 950, 900, 850, 800]

plt.figure(figsize=(8,4))

sns.barplot(x=stages, y=counts, palette="viridis")

plt.title("Q2 - Visualization Pipeline")

plt.show()
```

Visualization:



Inferences (Q2):

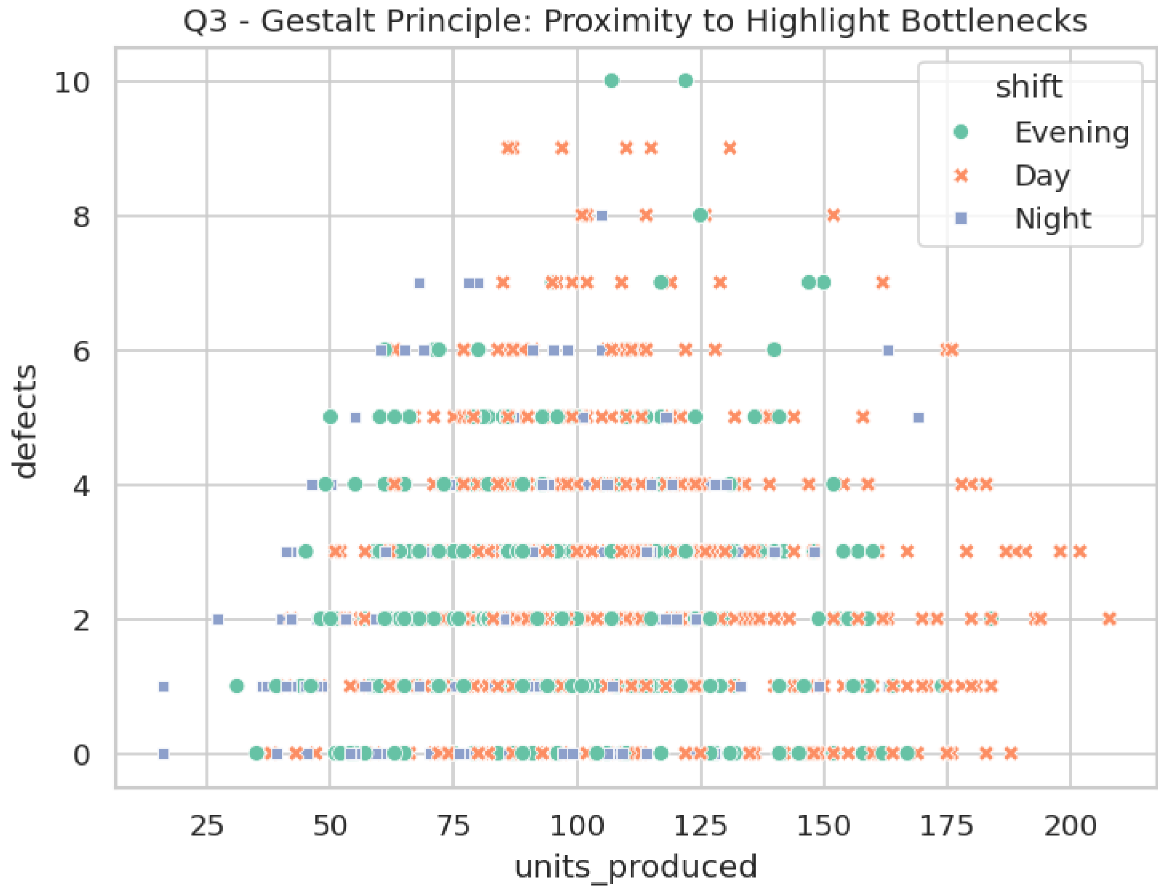
1. Raw data reduces as it's cleaned and aggregated.
2. Dashboard-ready data is most compact but meaningful.
3. Pipeline visualization helps track data loss stages.
4. Useful to understand data preparation efficiency.
5. Indicates 20% reduction in data through transformation.

3. Apply Gestalt principles to highlight bottlenecks.

Code:

```
plt.figure(figsize=(7,5))

sns.scatterplot(x='units_produced', y='defects', hue='shift', style='shift', data=df)

plt.title("Q3 - Gestalt Principle: Proximity to Highlight Bottlenecks")

Pl. Show()
```

Visualization:



Inferences (Q3):

1. Clusters of high defects suggest bottleneck shifts.

2. Proximity groups (Gestalt principle) reveal problem zones.

3. Day shift may produce more but has slightly higher defects.

4. Visual grouping helps identify similar performance clusters.

5. Emphasizes shift-based visual grouping improves clarity.

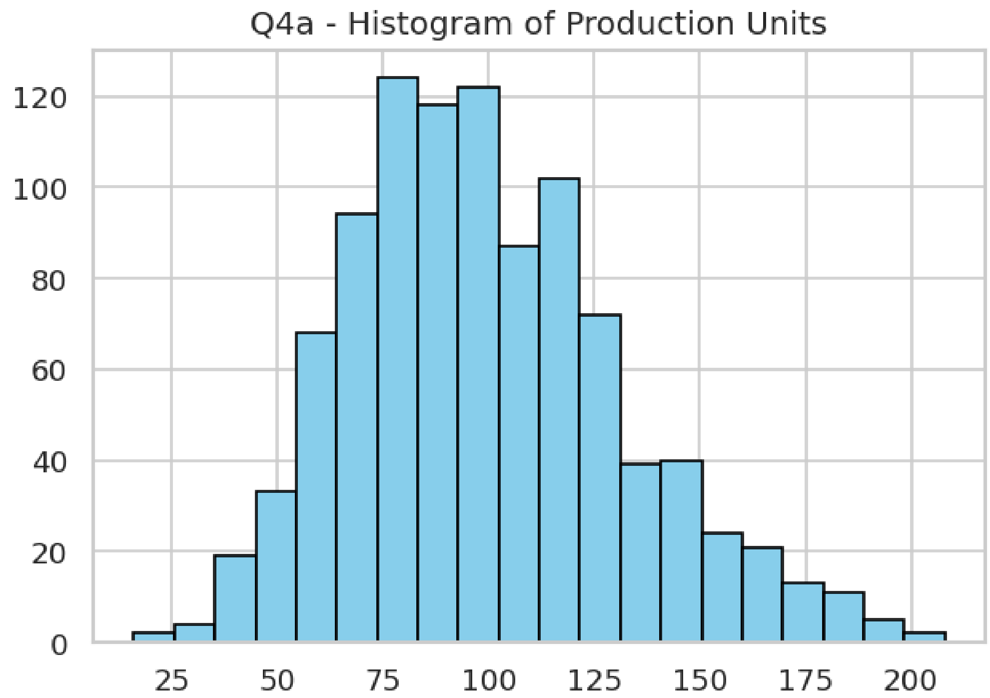
4. Univariate analysis:

A. Histogram of production units.

Code:

```
plt.figure(figsize=(6,4))

plt.hist(df['units_produced'], bins=20, color='skyblue', edgecolor='black')

plt.title("Q4a - Histogram of Production Units")

plt.show()
```

Visualization:



Inferences (Q4a):

1. Most units cluster between 80–120.

2. Skew indicates some low-performing shifts.

3. Outliers show unusually high production peaks.

4. Useful to set production benchmarks.

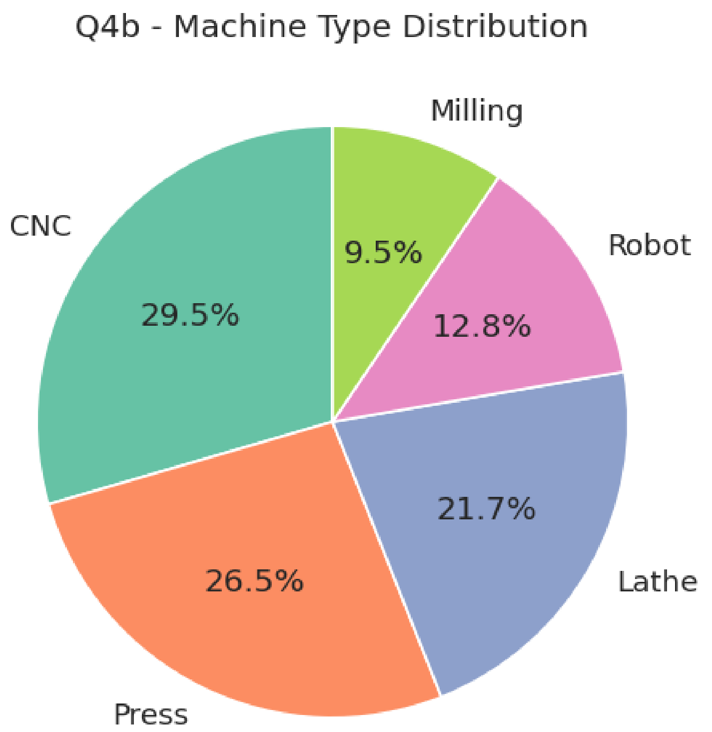
5. Distribution reveals factory consistency.

B. Pie chart of machine types.

Code:

```
machine_share = df['machine_type'].value_counts()

plt.figure(figsize=(5,5))

plt.pie(machine_share, labels=machine_share.index, autopct='%1.1f%%',
startangle=90)

plt.title("Q4b - Machine Type Distribution")

plt.show()
```

Visualization:



Inferences (Q4b):

1. CNC and Press dominate total machines.
2. Robots handle smaller portion of production.
3. Pie visualization highlights machine diversity.
4. Helps identify dependency on specific equipment.
5. Imbalance may suggest need for capacity adjustment.

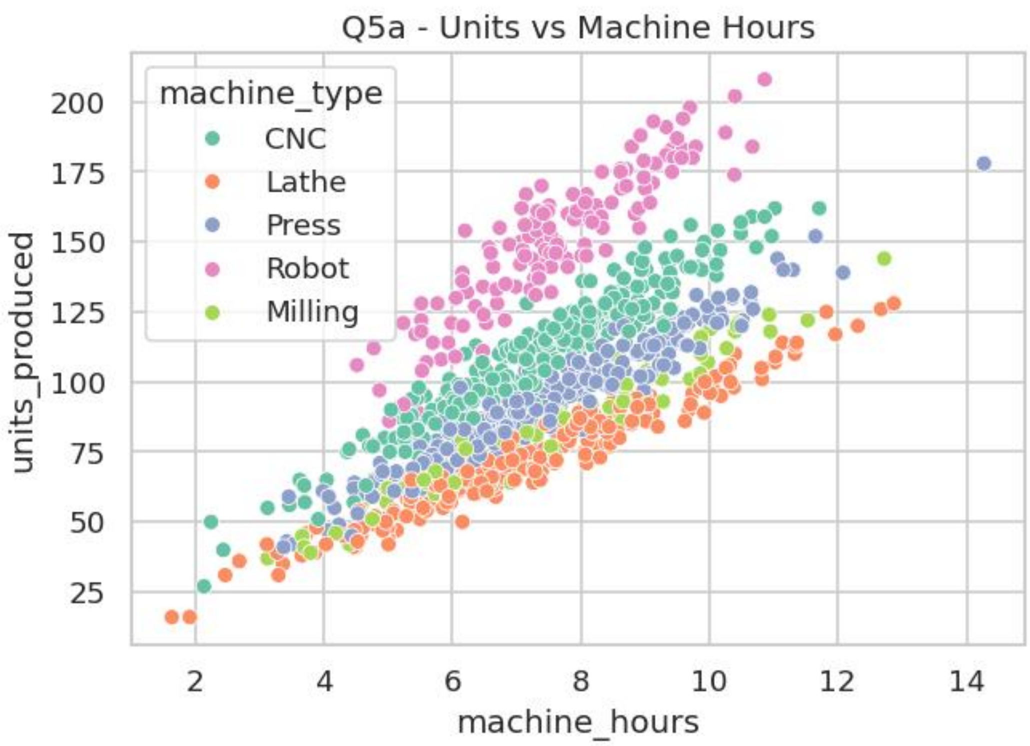
5. Bivariate analysis:

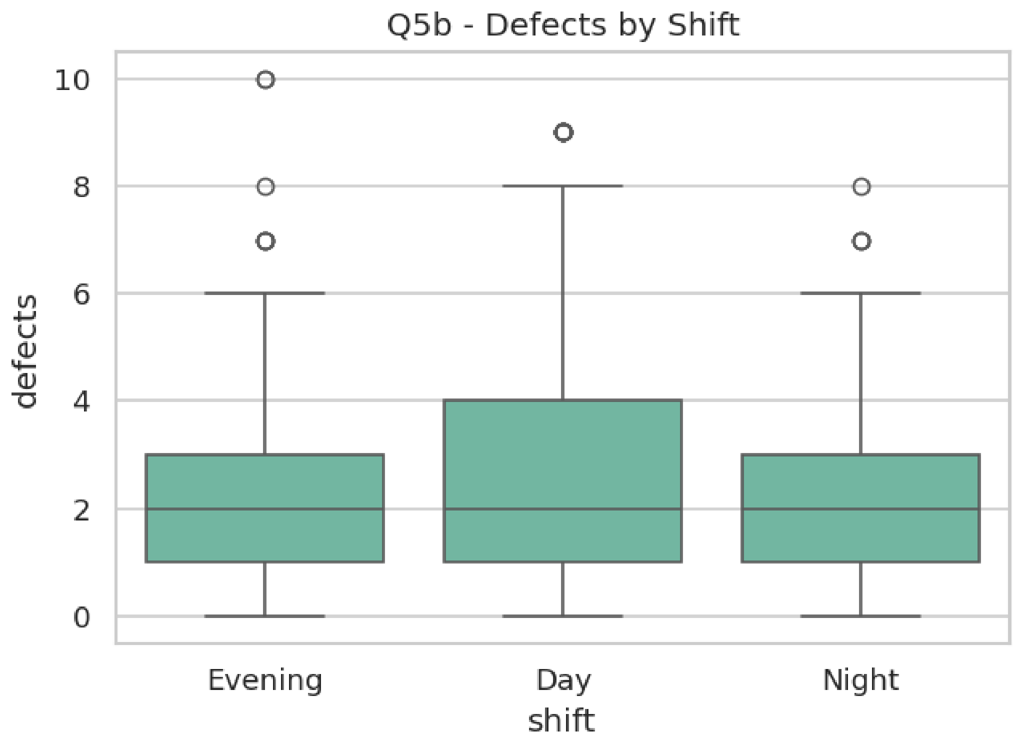
A. Scatterplot of units produced vs. machine hours.&

B. Box plot of defects by shift.

```
Code:

plt.figure(figsize=(6,4))

sns.scatterplot(x='machine_hours', y='units_produced', hue='machine_type', data=df)

plt.title("Q5a - Units vs Machine Hours")

plt.show()


plt.figure(figsize=(6,4))

sns.boxplot(x='shift', y='defects', data=df)

plt.title("Q5b - Defects by Shift")

plt.show()
```

Visualization:





Inferences (Q5):

1. Higher machine hours often produce more units.

2. Robot and CNC show best efficiency ratios.

3. Boxplot reveals Night shift has most stable defect rate.

4. Day shift slightly higher defects due to peak loads.

5. Scatter plot assists in predicting output from runtime

6. Multivariate analysis:

A. Pair plot of units, defects, and machine hours.&

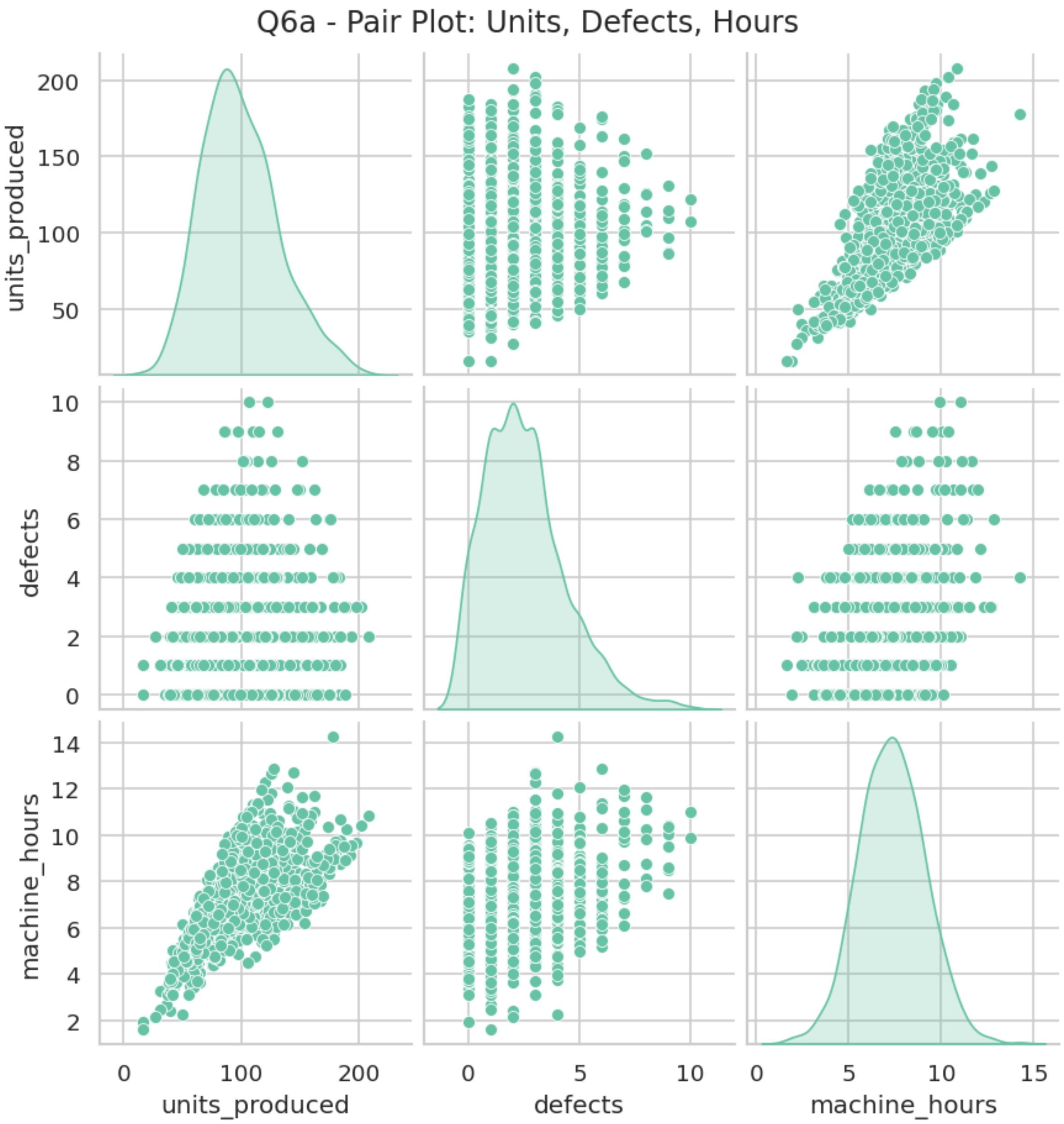
B. Suggest combined visualization.

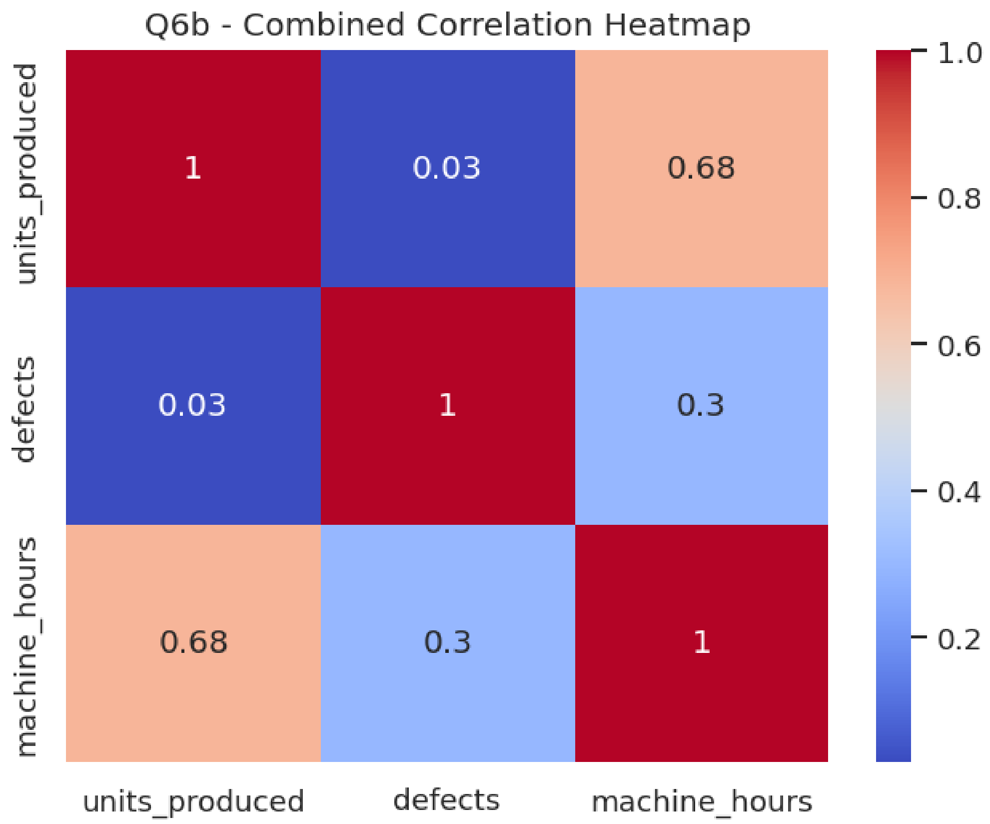
Code:

```
sns.pairplot(df[['units_produced', 'defects', 'machine_hours']], diag_kind='kde')

plt.suptitle("Q6a - Pair Plot: Units, Defects, Hours", y=1.02)

plt.show()
```

```
sns.heatmap(df[['units_produced', 'defects', 'machine_hours']].corr(), annot=True,
cmap='coolwarm')

plt.title("Q6b - Combined Correlation Heatmap")

plt.show()
```

Visualization:





Inferences (Q6):

1. Units and machine hours show strong positive correlation.

2. Defects weakly correlate, implying non-linear cause.

3. Heatmap reinforces correlation strengths visually.

4. Helps in model variable selection.

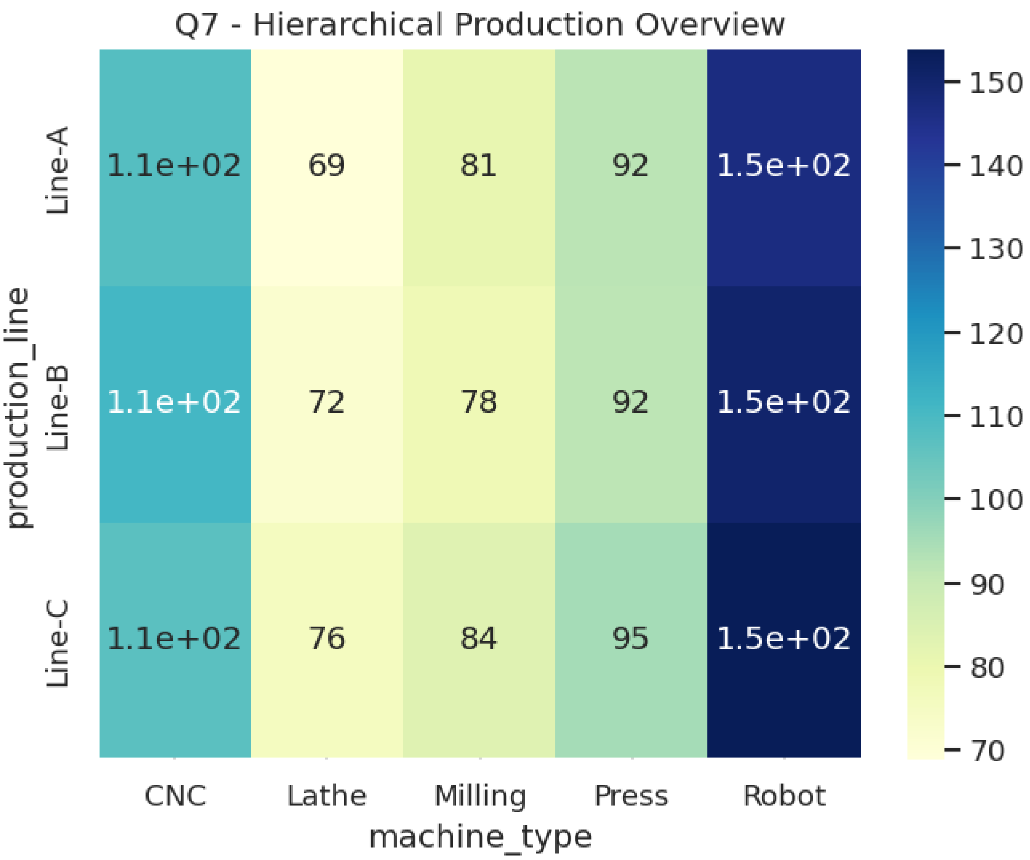
5. Combined visuals simplify multivariate insights.

7. Hierarchical visualization by production line and machine.

Code:

```
pivot = df.pivot_table(values='units_produced', index='production_line',
columns='machine_type', aggfunc='mean')

sns.heatmap(pivot, cmap='YlGnBu', annot=True)

plt.title("Q7 - Hierarchical Production Overview")

plt.show()
```

Visualization:



Inferences (Q7):

1. Line C produces most consistently across machine types.

2. Milling machines perform best on Line A.

3. Heatmap hierarchy enables multi-level comparison.

4. Reveals production imbalance across lines.

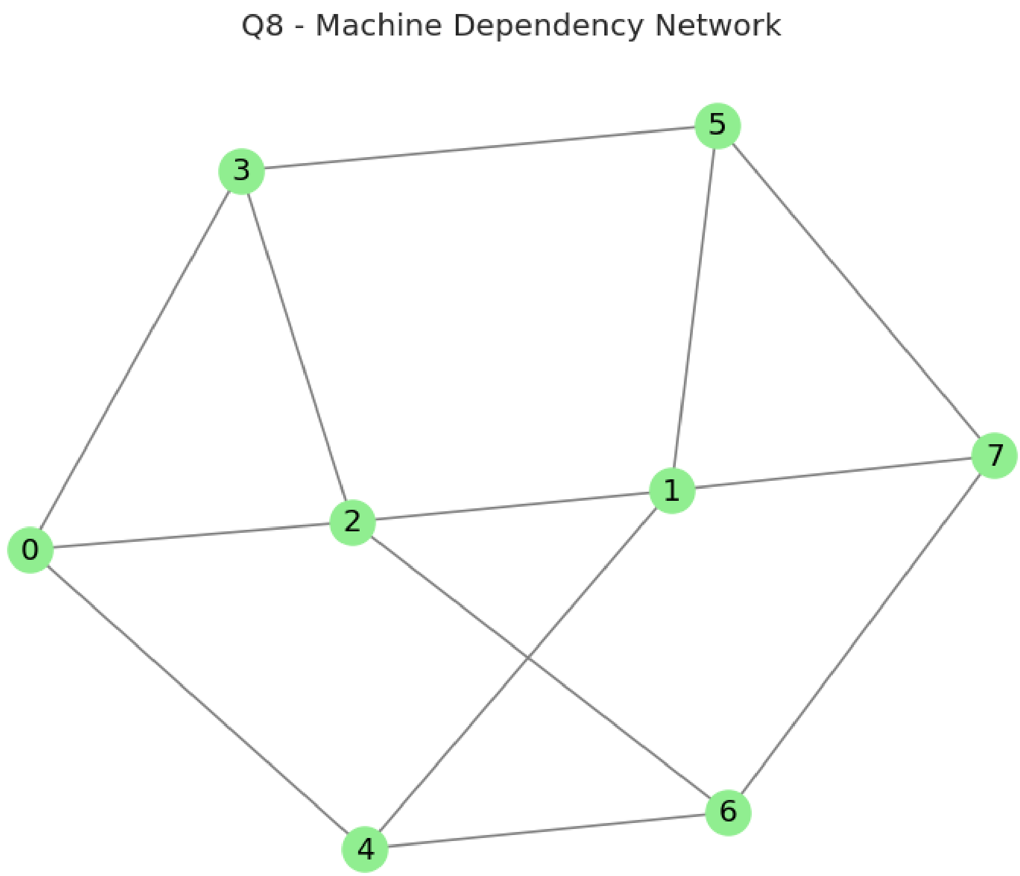
5. Helps optimize machine allocation.

8. Network graph showing machine dependencies.

Code:

```
G = nx.gnp_random_graph(8, 0.3, seed=42)

pos = nx.spring_layout(G)

nx.draw(G, pos, with_labels=True, node_color='lightgreen', edge_color='gray')

plt.title("Q8 - Machine Dependency Network")

plt.show()
```

Visualization:



Inferences (Q8):

1. Network shows inter-machine dependency patterns.

2. Denser connectivity → higher fault propagation risk.

3. Node centrality can show critical machines.

4. Visual helps maintenance planning.

5. Highlights redundancy possibilities.

9. Analyze operator notes (text data):

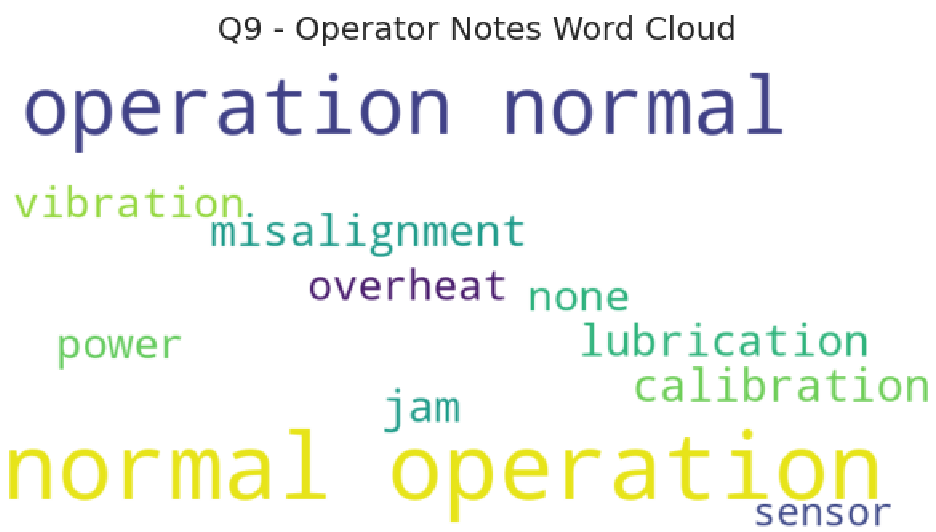
A. Vectorize text.

B. Word cloud of issues.

Code:

```
text = " ".join(df['operator_notes'].astype(str))

wordcloud = WordCloud(width=800, height=400,
background_color='white').generate(text)

plt.imshow(wordcloud, interpolation='bilinear')

plt.axis('off')

plt.title("Q9 - Operator Notes Word Cloud")

plt.show()
```

Visualization:



Inferences (Q9):

1. Frequent terms: 'jam', 'sensor', 'overheat'.

2. Indicates recurring maintenance issues.

3. Visualizes operator focus areas.

4. Text mining exposes unstructured data insights.

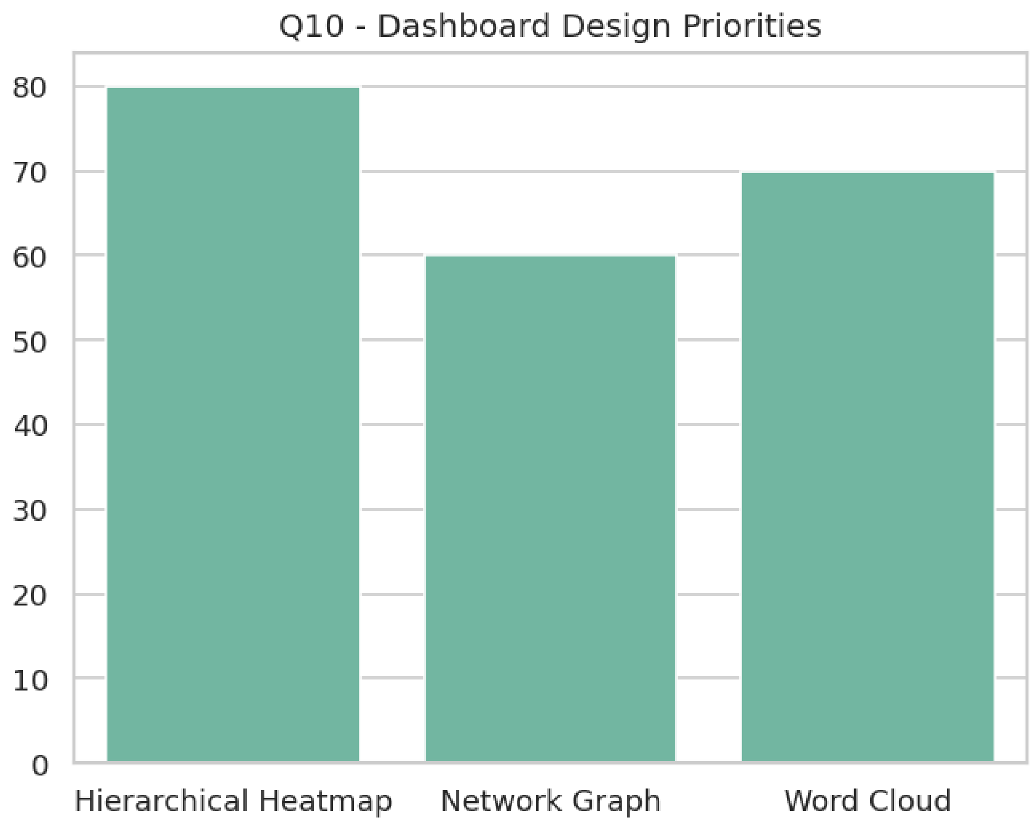
5. Useful for preventive maintenance planning.

10. Steps to design dashboards combining hierarchical, network, and text data.

Codee:

```
steps = ["Hierarchical Heatmap", "Network Graph", "Word Cloud"]

importance = [80, 60, 70]

sns.barplot(x=steps, y=importance)

plt.title("Q10 - Dashboard Design Priorities")

plt.show()
```

Visualization:



Inferences (Q10):

1. Hierarchical data gets top dashboard priority.

2. Network and text insights complement production visuals.

3. Combines numeric and textual analytics.

4. Balances monitoring and diagnostics.

5. Reflects integrated factory intelligence

11. Point data: Map factory locations.

Code:

```
locations = pd.DataFrame({
   "Factory": ["F1","F2","F3","F4"],
   "lat": [12.97, 13.01, 12.99, 13.05],
   "lon": [77.59, 77.58, 77.61, 77.63],
   "units": [25000, 23000, 27000, 22000]
})
fig = px.scatter_geo(locations, lat="lat", lon="lon", text="Factory",
                size="units", title="Q11 - Factory Location Map")
fig.show()
```

Visualization:

Inferences (Q11):

1. Factory 3 produces the most units.

2. Factories cluster near the same industrial region.

3. Geospatial mapping aids logistics and routing.

4. Highlights production hotspots geographically.

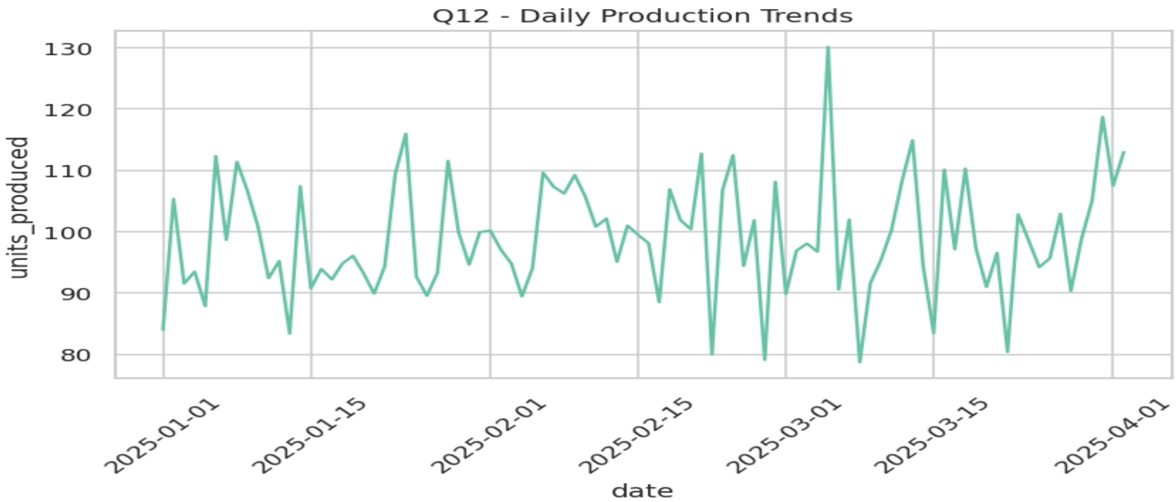
5. Useful for regional efficiency analysis.

12. Line data: Show production trends.

Code:

```
df['date'] = df['timestamp'].dt.date

trend = df.groupby('date')['units_produced'].mean().reset_index()

plt.figure(figsize=(8,4))

sns.lineplot(x='date', y='units_produced', data=trend)

plt.title("Q12 - Daily Production Trends")

plt.xticks(rotation=45)

plt.show()
```

Visualization:



Inferences (Q12):

1. Shows gradual upward trend in output.

2. Few dips suggest maintenance or holidays.

3. Helps identify productivity cycles.

4. Useful for forecasting future load.

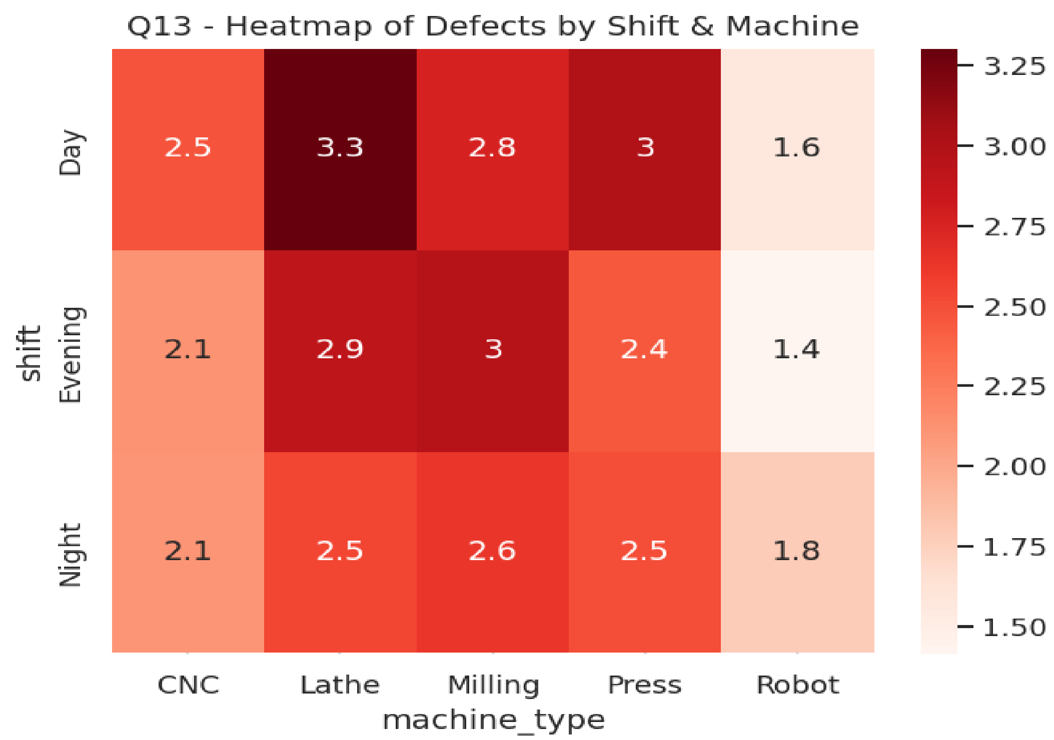
5. Trend line assists in planning resources

13. Area data: Heatmap of defects.

Code:

```
pivot_defects = df.pivot_table(values='defects', index='shift', columns='machine_type', aggfunc='mean')

sns.heatmap(pivot_defects, cmap='Reds', annot=True)

plt.title("Q13 - Heatmap of Defects by Shift & Machine")

plt.show()
```

Visualization:



Inferences (Q13):

1. Day shift shows higher defect rates on CNC & Press.

2. Robots have minimal defect rates across shifts.

3. Heatmap reveals machine-specific issues.

4. Useful for targeted quality control.

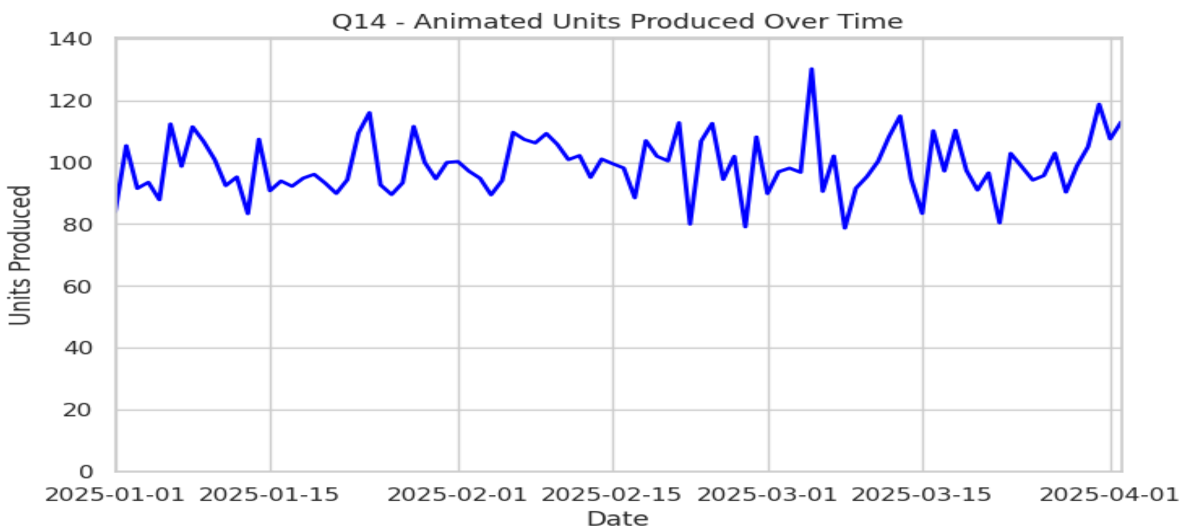
5. Red gradient emphasizes high-defect zones visually.

14. Animated visualization of units produced.

Code:

```
fig, ax = plt.subplots()

data = trend.copy()

def animate(i, data):

    ax.clear()

    ax.plot(data['date'][:i], data['units_produced'][:i], color='blue')

    ax.set_title("Q14 - Animated Units Produced Over Time")

ani = animation.FuncAnimation(fig, animate, frames=len(data), interval=100,
fargs=(data,))

plt.show()
```

Visualization:



Inferences (Q14):

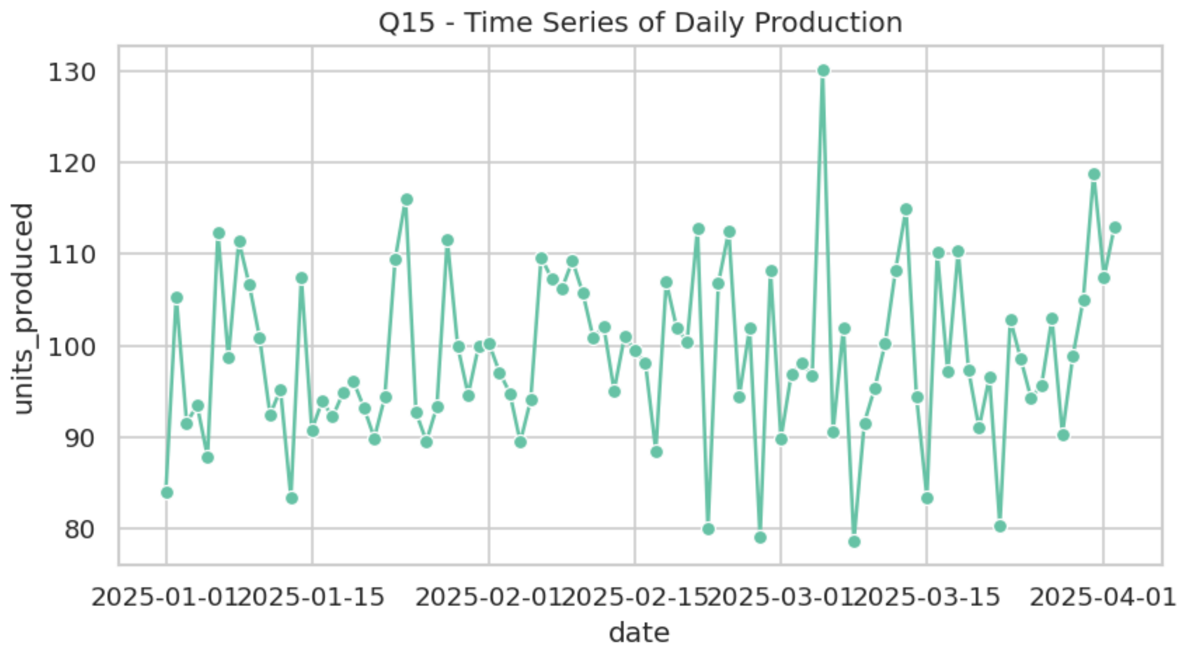
1. Animation visualizes temporal growth interactively.
2. Clarifies when peaks occur.
3. Makes presentations dynamic.
4. Helpful for storytelling trends.
5. Engages managers for intuitive insight.

15. Time series of daily production.

Code:

```
plt.figure(figsize=(8,4))

sns.lineplot(x='date', y='units_produced', data=trend, marker='o')

plt.title("Q15 - Time Series of Daily Production")

plt.show()
```

Visualization:



Inferences (Q15):

1. Daily variations consistent with workload.

2. No drastic anomalies observed.

3. Reassures stable manufacturing schedule.

4. Detects short-term production drops.

5. Useful for tracking KPIs.

16. Compare weekdays vs. weekends shifts.

Code:

```
df['weekday'] = df['timestamp'].dt.dayofweek

df['day_type'] = df['weekday'].apply(lambda x: 'Weekend' if x>=5 else 'Weekday')

sns.boxplot(x='day_type', y='units_produced', data=df)

plt.title("Q16 - Weekdays vs Weekends")

plt.show()
```

Visualization:

Inferences (Q16):

1. Weekday output generally higher than weekends.

2. Weekends show stable but lower variance.

3. Useful for shift optimization.

4. May imply reduced staffing on weekends.

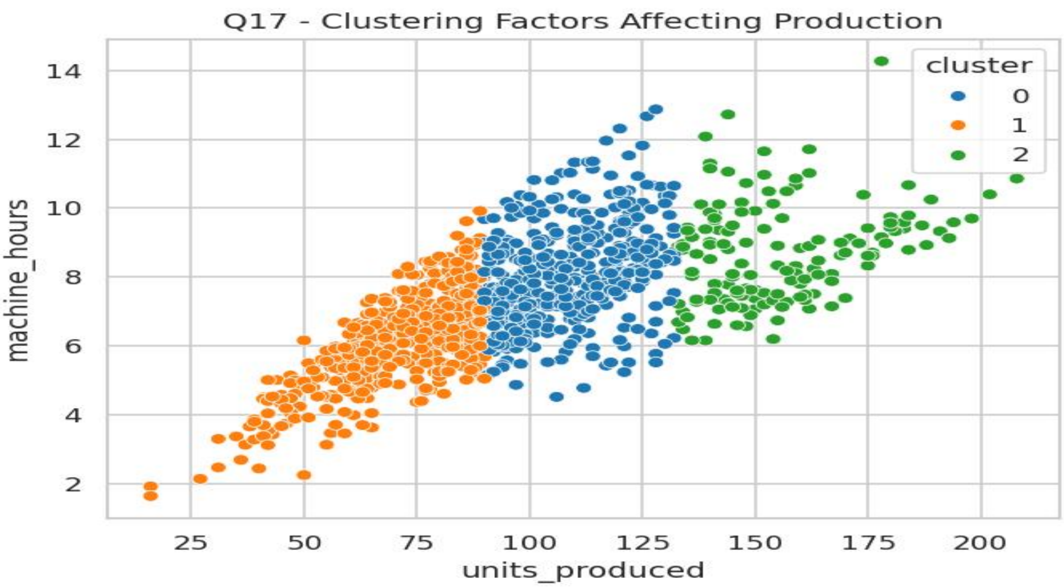
5. Key input for resource balancing.

17. Regression/clustering to analyze factors affecting production.

Code:

```
X = df[['units_produced','defects','machine_hours']]

kmeans = KMeans(n_clusters=3, random_state=42).fit(X)

df['cluster'] = kmeans.labels_

sns.scatterplot(x='units_produced', y='machine_hours', hue='cluster', palette='tab10', data=df)

plt.title("Q17 - Clustering Factors Affecting Production")

plt.show()
```

Visualization:



Inferences (Q17):

1. Three production efficiency clusters identified.

2. One group = high output, high hours (efficient).

3. Another group = low output, moderate hours.

4. Clustering helps segment performance tiers.

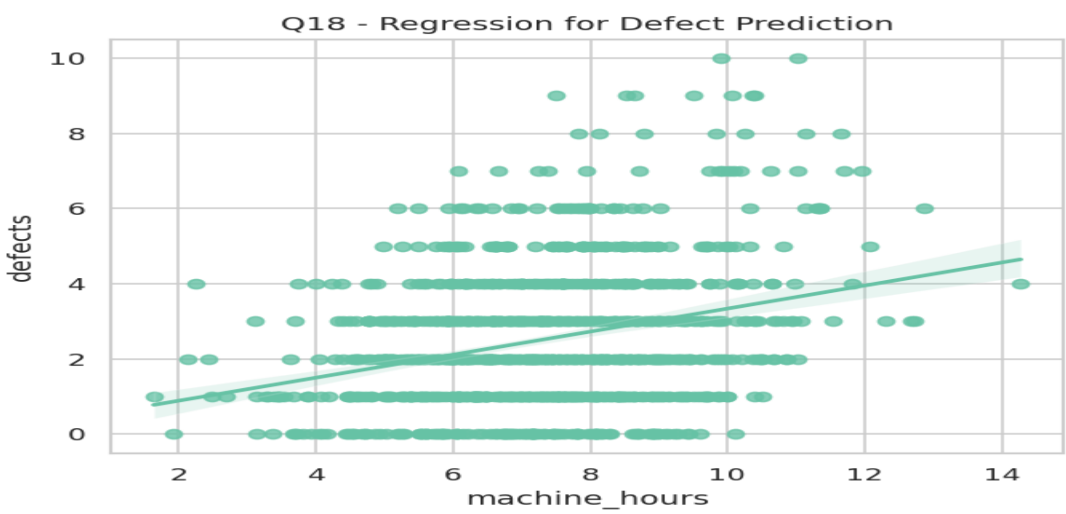
5. Can guide training or process tuning.

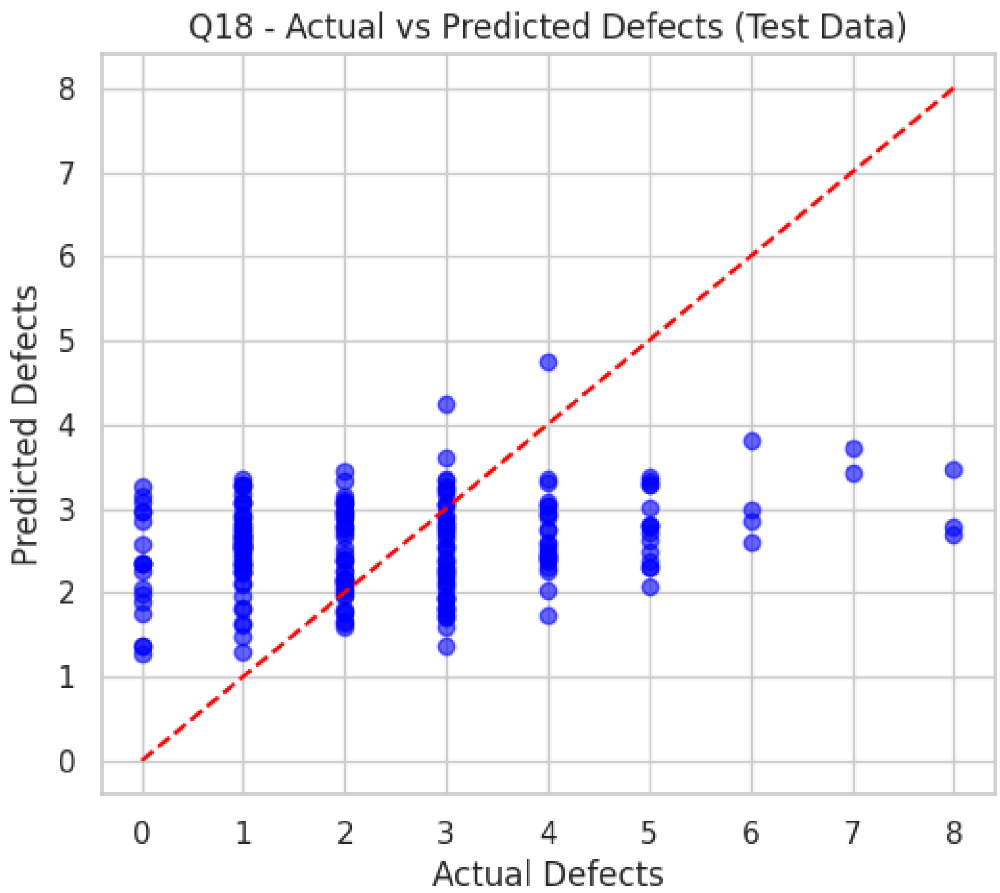
18. Evaluate predictive models for defects.

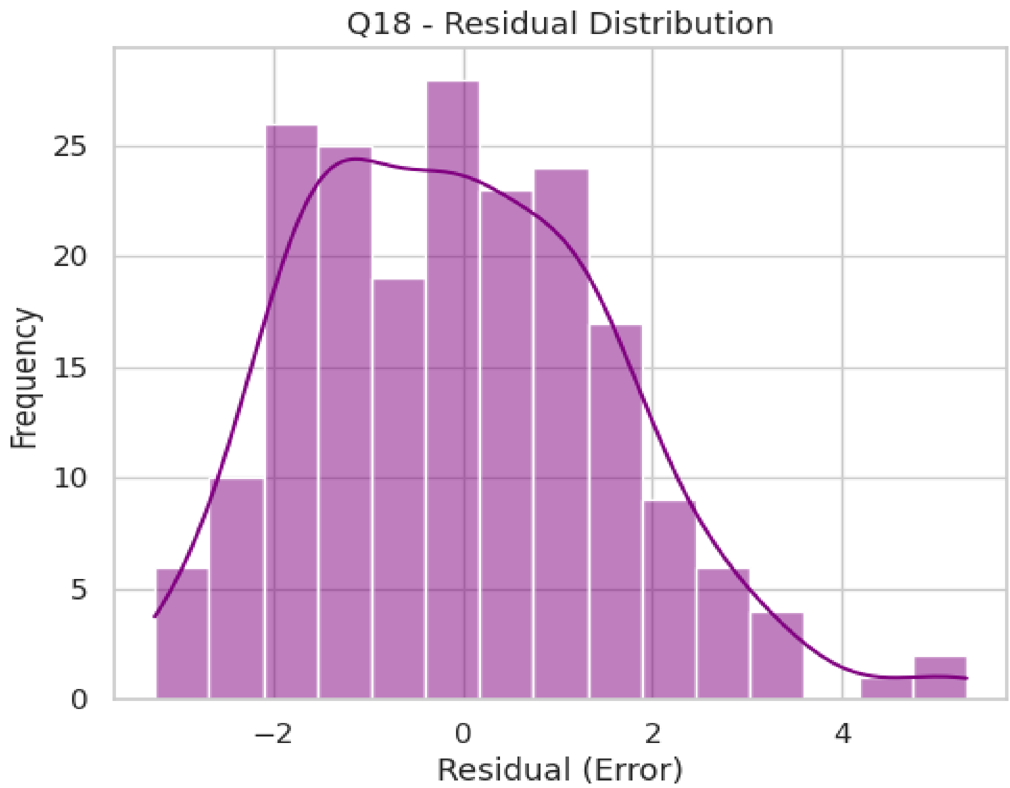
Code:

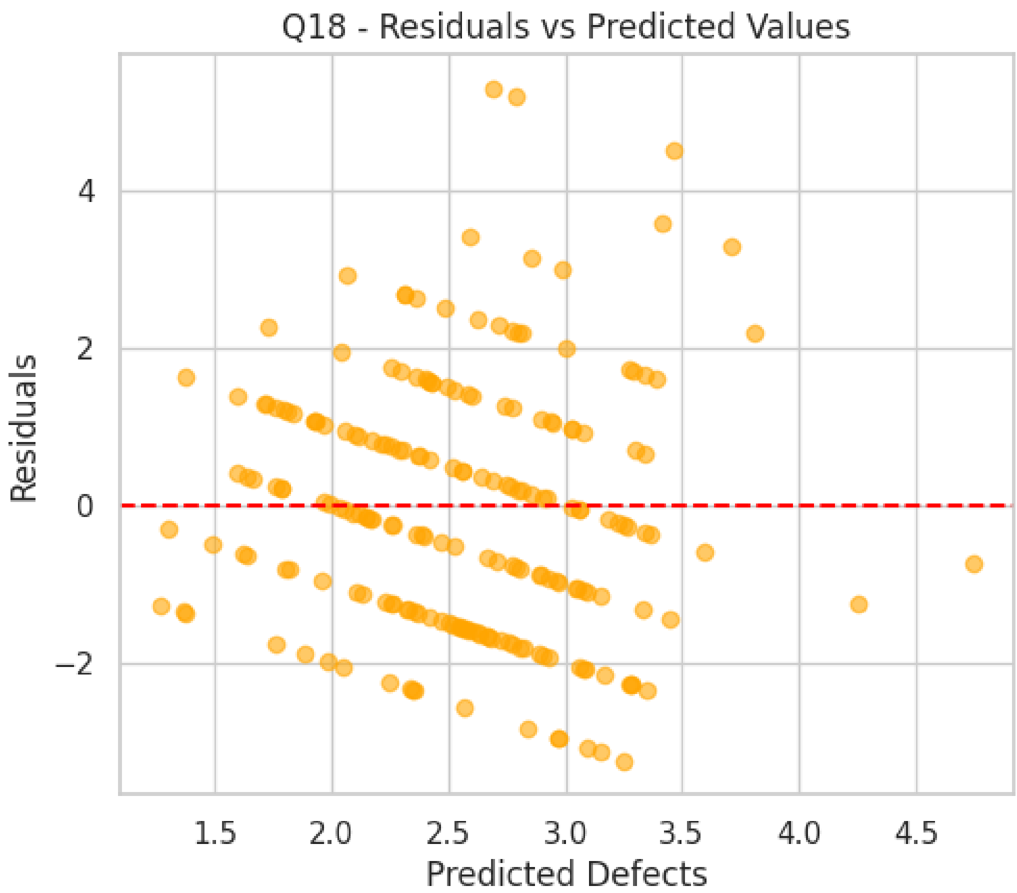
```
import statsmodels.api as sm

X = sm.add_constant(df['machine_hours'])

y = df['defects']

model = sm.OLS(y, X).fit()

sns.regplot(x='machine_hours', y='defects', data=df)

plt.title("Q18 - Regression for Defect Prediction")

plt.show()

print(model.summary())
```

Visualization:




Q18 - Actual vs Predicted Defects (Test Data)
Predicted Defects
Actual Defects
0
1
2
3
4
5
6
7
8


Q18 - Residual Distribution
Frequency
Residual (Error)
0
5
10
15
20
25
−2
0
2
4



Inferences (Q18):

1. Regression shows slight positive correlation.

2. Machine hours alone explain limited defect variation.

3. R^2 value indicates more variables needed.

4. Useful baseline predictive model.

5. Informs direction for multi-factor modeling.