4.Data Cleaning + Statistical Modeling in pandas

(i)messy_data.csv

```
%%writefile messy_data.csv
Name,Age,Salary,Experience,Hired
Alice,29,50000,5,Yes
Bob,,55000,6,No
Charlie,35,error,8,Yes
David,45,70000,,Yes
Eve,50,85000,10,No
```

```
Writing messy_data.csv
```

Gemini

```
import pandas as pd

df = pd.read_csv('messy_data.csv')
display(df)
```

	Name	Age	Salary	Experience	Hired
0	Alice	29.0	50000	5.0	Yes
1	Bob	NaN	55000	6.0	No
2	Charlie	35.0	error	8.0	Yes
3	David	45.0	70000	NaN	Yes
4	Eve	50.0	85000	10.0	No

Next steps: Generate code with df | View recommended plots | New interactive sheet

(ii)data_prep.ipynb

```
import pandas as pd
import numpy as np

# Load the data
df = pd.read_csv("messy_data.csv")

# 1. Convert Salary to numeric (handle 'error')
df["Salary"] = pd.to_numeric(df["Salary"], errors="coerce")

# 2. Fill missing Age with mean
df["Age"].fillna(df["Age"].mean(), inplace=True)

# 3. Fill missing Experience with median
```

```
df["Experience"].fillna(df["Experience"].median(), inplace=True)

# 4. Fill missing Salary with median
df["Salary"].fillna(df["Salary"].median(), inplace=True)

# 5. Convert 'Hired' column to numeric
df["Hired"] = df["Hired"].map({"Yes": 1, "No": 0})

# 6. Feature engineering: Age × Experience
df["Age_Exp"] = df["Age"] * df["Experience"]

# 7. Drop Name column (not useful for modeling)
df.drop("Name", axis=1, inplace=True)

# Save cleaned data
df.to_csv("cleaned_data.csv", index=False)

# Show cleaned DataFrame
print(df)
```

```
      Age   Salary  Experience  Hired  Age_Exp
0  29.00  50000.0         5.0      1    145.0
1  39.75  55000.0         6.0      0    238.5
2  35.00  62500.0         8.0      1    280.0
3  45.00  70000.0         7.0      1    315.0
4  50.00  85000.0        10.0      0    500.0
/tmp/ipython-input-577083859.py:11: FutureWarning: A value is trying to be set o
The behavior will change in pandas 3.0. This inplace method will never work beca

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.met


  df["Age"].fillna(df["Age"].mean(), inplace=True)
/tmp/ipython-input-577083859.py:14: FutureWarning: A value is trying to be set o
The behavior will change in pandas 3.0. This inplace method will never work beca

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.met


  df["Experience"].fillna(df["Experience"].median(), inplace=True)
/tmp/ipython-input-577083859.py:17: FutureWarning: A value is trying to be set o
The behavior will change in pandas 3.0. This inplace method will never work beca

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.met


  df["Salary"].fillna(df["Salary"].median(), inplace=True)
```

(iii)modeling.ipynb

```
import pandas as pd
import statsmodels.api as sm

# Load cleaned data
df = pd.read_csv("cleaned_data.csv")
```

```
# Define X (features) and y (target)
X = df[["Age", "Salary", "Experience", "Age_Exp"]]
y = df["Hired"]

# Add constant for intercept
X = sm.add_constant(X)

# Fit logistic regression model
model = sm.Logit(y, X).fit()

# Show model summary
print(model.summary())

# Predict
df["Predicted_Prob"] = model.predict(X)
df["Predicted_Label"] = (df["Predicted_Prob"] > 0.5).astype(int)

# Accuracy
accuracy = (df["Predicted_Label"] == y).mean()
print(f"\nAccuracy: {accuracy:.2f}")

# Display predictions
print(df[["Hired", "Predicted_Prob", "Predicted_Label"]])
```

```
Warning: Maximum number of iterations has been exceeded.
         Current function value: 0.009979
         Iterations: 35
                           Logit Regression Results
==============================================================================
Dep. Variable:                  Hired   No. Observations:                    5
Model:                          Logit   Df Residuals:                        0
Method:                           MLE   Df Model:                            4
Date:                Sun, 03 Aug 2025   Pseudo R-squ.:                  0.9852
Time:                        12:08:57   Log-Likelihood:               -0.049893
converged:                      False   LL-Null:                        -3.3651
Covariance Type:            nonrobust   LLR p-value:                    0.1568
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const        -12.9979        nan        nan        nan         nan         nan
Age           -2.6687        nan        nan        nan         nan         nan
Salary         0.0046        nan        nan        nan         nan         nan
Experience   -13.1591        nan        nan        nan         nan         nan
Age_Exp       -0.2369        nan        nan        nan         nan         nan
==============================================================================

Possibly complete quasi-separation: A fraction 0.40 of observations can be
perfectly predicted. This might indicate that there is complete
quasi-separation. In this case some parameters will not be identified.

Accuracy: 1.00
   Hired  Predicted_Prob  Predicted_Label
0      1        1.000000                1
1      0        0.047870                0
2      1        0.999606                1
3      1        1.000000                1
4      0        0.000444                0
```

```
/usr/local/lib/python3.11/dist-packages/statsmodels/base/model.py:607: Convergen
  warnings.warn("Maximum Likelihood optimization failed to "
```