# Q1. Difference between Ordinal Encoding and Label Encoding

| Aspect | Ordinal Encoding | Label Encoding |
|---|---|---|
| Definition | Assigns integers based on **intrinsic order** of categories | Assigns integers arbitrarily to categories (no order implied) |
| When to use | When the categorical variable **has a natural order** (e.g., low < medium < high) | When categorical variable **has no natural order** (nominal) |
| Example | Education: High School=0, Bachelor=1, Master=2, PhD=3 | Color: Red=0, Blue=1, Green=2 |

**Key point:**

- **Ordinal encoding** preserves the order
- **Label encoding** just converts categories to numbers

---

# Q2. Target Guided Ordinal Encoding

**Definition:**
Target Guided Ordinal Encoding maps categories of a categorical feature to **numbers based on their relationship with the target variable** (usually the mean of the target for each category).

**Example Use Case:**

- Dataset: Predicting loan default (target = 0/1)
- Feature: `Employment Type` = [Salaried, Self-Employed, Unemployed]
- Calculate mean default rate for each category:

| Employment Type | Mean Default Rate | Encoded Value |
|---|---|---|

| | | |
|---|---|---|
| Unemployed | 0.8 | 3 |
| Self-Employed | 0.4 | 2 |
| Salaried | 0.1 | 1 |

```
import pandas as pd

df = pd.DataFrame({
    'Employment':
['Salaried','Self-Employed','Unemployed','Salaried','Unemployed'],
    'Default':[0,1,1,0,1]
})

# Mean target encoding
encoding_map =
df.groupby('Employment')['Default'].mean().sort_values().rank().to_dic
t()
df['Employment_encoded'] = df['Employment'].map(encoding_map)
```

> **Benefit:** Captures the relationship between the categorical feature and target variable, often improving model performance.

---

# Q3. Covariance

**Definition:**
Covariance measures **how two variables change together**.

- Positive covariance → variables increase together
- Negative covariance → one increases while the other decreases
- Zero covariance → variables are independent

**Formula:**

Cov(X,Y)=∑(Xi−X⁻)(Yi−Y⁻)n−1\text{Cov}(X,Y) = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{n-1}Cov(X,Y)=n−1∑(Xi−X⁻)(Yi−Y⁻)

**Importance:**

- Helps identify relationships between variables
- Used in **PCA, portfolio optimization, and statistical analysis**

---

## Q4. Label Encoding using scikit-learn

Variables:

- Color = {red, green, blue}
- Size = {small, medium, large}
- Material = {wood, metal, plastic}

```
from sklearn.preprocessing import LabelEncoder
import pandas as pd

df = pd.DataFrame({
    'Color':['red','green','blue','green','red'],
    'Size':['small','medium','large','medium','small'],
    'Material':['wood','metal','plastic','wood','metal']
})

le = LabelEncoder()
for col in df.columns:
    df[col+'_encoded'] = le.fit_transform(df[col])

print(df)
```

**Output Explanation:**

- Each category is assigned an integer **arbitrarily** (no order implied)
- Example: `Color` → blue=0, green=1, red=2
- Now dataset is **ML-ready**

---

## Q5. Covariance Matrix Example

Variables: Age, Income, Education level

```
import numpy as np

data = np.array([
    [25, 50000, 12],
    [30, 60000, 16],
    [22, 45000, 10],
    [28, 52000, 14]
])

cov_matrix = np.cov(data.T)
print(cov_matrix)
```

**Interpretation:**

- `cov_matrix[i][j]` shows covariance between variable i and j
- Positive values → increase together
- Negative values → inversely related
- Diagonal → variance of each variable

---

## Q6. Encoding categorical variables in ML project

| Variable | Suggested Encoding | Reason |
|---|---|---|
| Gender (Male/Female) | Label encoding (0/1) | Binary variable |
| Education Level (High School/Bachelor/Master/PhD) | Ordinal encoding | Has natural order |

| | | |
|---|---|---|
| Employment Status (Unemployed/Part-Time/Full-Time) | Ordinal encoding (0,1,2) or One-hot | Can be ordered by hours worked or leave as nominal for tree-based models |

---

# Q7. Covariance for continuous and categorical variables

Dataset:

- Continuous: Temperature, Humidity
- Categorical: Weather Condition (Sunny/Cloudy/Rainy), Wind Direction (N/S/E/W)

**Steps:**

1. Encode categorical variables (e.g., Label Encoding)
2. Calculate covariance

```
import pandas as pd
import numpy as np

df = pd.DataFrame({
    'Temperature':[30,25,28,27,32],
    'Humidity':[70,65,60,75,80],
    'Weather':['Sunny','Cloudy','Rainy','Sunny','Rainy'],
    'Wind':['N','S','E','W','N']
})

# Label encode categorical variables
for col in ['Weather','Wind']:
    df[col] = LabelEncoder().fit_transform(df[col])

cov_matrix = df.cov()
print(cov_matrix)
```

**Interpretation:**

- Covariance between `Temperature` and `Humidity` → positive if hotter days are more humid
- Covariance between continuous and encoded categorical → gives rough relationship, but **use with caution** because label encoding can introduce artificial order

---

## ✅ Summary

- **Label Encoding:** Arbitrary integer assignment
- **Ordinal Encoding:** Integer assignment preserving natural order
- **Target Guided Encoding:** Encodes based on relationship with target
- **Covariance:** Measures co-variation between variables
- **Practical Use:** Preprocessing categorical data, understanding relationships, PCA, feature selection

---