Q1: Data encoding: Data encoding is the process of converting categorical data into numerical format, which can be understood and processed by machine learning algorithms. It is useful in data science because many machine learning algorithms require numerical input, and encoding categorical variables allows us to represent them in a format suitable for analysis.

Q2: Nominal encoding: Nominal encoding, also known as label encoding, assigns a unique integer value to each category in a categorical variable. It is suitable for variables where there is no inherent order or hierarchy among the categories.
Example: Suppose you have a dataset with a "color" variable containing categories like "red," "green," and "blue." Nominal encoding assigns integer values to each category, such as 0 for "red," 1 for "green," and 2 for "blue."

Q3: When is nominal encoding preferred over one-hot encoding?
Nominal encoding is preferred over one-hot encoding when the categorical variable has a large number of unique categories, as one-hot encoding would create a large number of new columns, leading to the curse of dimensionality and potentially overfitting the model.
Example: In a dataset with a "country" variable containing hundreds of unique country names, nominal encoding can efficiently represent the categories using integer values without expanding the feature space excessively.

Q4: Choice of encoding technique for a dataset with 5 unique values:
For a dataset with a categorical variable containing 5 unique values, nominal encoding is typically used. This is because nominal encoding assigns a single integer value to each category, making it more efficient and compact compared to one-hot encoding, which would create 5 new binary columns.
Q5: Number of new columns created with nominal encoding: If you have two categorical columns in a dataset with 5 unique values each, and you use nominal encoding, each column will be replaced with a single numerical column. Therefore, using nominal encoding would create 2 new columns in total.

Q6: Choice of encoding technique for a dataset containing information about animals: For a dataset containing categorical variables like "species" and "habitat," where there is no inherent order or hierarchy among the categories, nominal encoding (label encoding) would be suitable. This is because nominal encoding can efficiently represent the categories using integer values, allowing machine learning algorithms to process the data effectively.

Q7: Encoding technique(s) for predicting customer churn: For the dataset containing categorical variables like "gender" and "contract type," nominal encoding can be used. Here's a step-by-step explanation of how to implement the encoding:

1. **Identify categorical variables:** Determine which columns in the dataset contain categorical data. In this case, "gender" and "contract type" are categorical variables.
2. **Apply nominal encoding:** Use a label encoder from a library like scikit-learn to encode the categorical variables into numerical format. Each unique category will be assigned a unique integer value.

# Example dataset with categorical variables data = pd.DataFrame({ 'gender': ['male', 'female', 'male', 'female'], 'contract_type': ['monthly', 'annual', 'monthly', 'annual'] }) # Apply nominal encoding label_encoder = LabelEncoder() data['gender_encoded'] = label_encoder.fit_transform(data['gender']) data['contract_type_encoded'] = label_encoder.fit_transform(data['contract_type']) print(data)

This will create new columns "gender_encoded" and "contract_type_encoded" containing the encoded numerical values for the categorical variables. Now, the dataset is ready for use in machine learning algorithms that require numerical input.