Custom DataFrame Transformation

Implement a custom DataFrame transformation function using `apply` and `groupby`.

In [33]:

```
import pandas as pd
data = {
    'Region': ['North', 'North', 'North', 'South', 'South', 'South'],
    'Sales': [100, 200, 300, 400, 500, 600]
}

df = pd.DataFrame(data)

print("Original DataFrame:")
print(df)
```

```
Original DataFrame:
  Region  Sales
0  North    100
1  North    200
2  North    300
3  South    400
4  South    500
5  South    600
```

Custom Transformation Function

In [34]:

```
def transform_sales_data(df):
    grouped_df = df.groupby('Region')
    transformed_df = grouped_df['Sales'].apply(lambda x: pd.Series({
        'Total Sales': x.sum(),
        'Average Sales': x.mean(),
        'Sales Variance': x.var()
    }))

    return transformed_df
```

Applying the Transformation Function

In [35]:

```
transformed_df = transform_sales_data(df)

print("\nTransformed DataFrame:")
print(transformed_df)
```

```
Transformed DataFrame:
Region
North   Total Sales         600.0
        Average Sales       200.0
        Sales Variance    10000.0
South   Total Sales        1500.0
        Average Sales       500.0
        Sales Variance    10000.0
Name: Sales, dtype: float64
```

```
In [ ]:
```