Exercise Lab 2

Mina Moeini
Mina_moeini@sfu.ca

September 20, 2024

1 Basic Syntax

The `IF-THEN/ELSE` statement in SAS is used to conditionally execute commands or manipulate data based on a logical condition. Here's the basic syntax:

```
IF condition THEN action;
```

condition: The logical condition to be evaluated by SAS.
action: The command to be executed if the condition is true.

2 1. Simple `IF-THEN` Statement

A basic conditional statement that executes a single action when the condition is true.

```
IF Age >= 18 THEN Status = 'Adult';
```

In this case, if the value of `Age` is greater than or equal to 18, then the variable `Status` will be assigned the value 'Adult'.

3 2. `IF-THEN-DO` for Multiple Actions

To execute multiple actions for a single condition, you can use the `DO` block:

```
IF Age >= 18 THEN DO;
    Status = 'Adult';
    Voting_Eligible = 'Yes';
END;
```

When `Age` is greater than or equal to 18, both `Status` and `Voting_Eligible` are updated.

4 3. IF-THEN-ELSE Statement

You can add an ELSE statement to execute an action when the IF condition is false.

```
IF Age >= 18 THEN Status = 'Adult';
ELSE Status = 'Minor';
```

If the condition Age >= 18 is false, then Status is set to 'Minor'.

5 4. Multiple Conditions Using ELSE IF

You can chain multiple conditions using ELSE IF:

```
IF Age >= 65 THEN Status = 'Senior';
ELSE IF Age >= 18 THEN Status = 'Adult';
ELSE Status = 'Minor';
```

If Age is greater than or equal to 65, Status is set to 'Senior'. If Age is between 18 and 64, Status is 'Adult'. Otherwise, Status is 'Minor'.

6 5. IF-THEN-ELSE with Multiple Conditions (AND/OR)

You can combine multiple conditions using logical operators such as AND and OR:

```
IF Age >= 18 AND Employed = 1 THEN Eligible_For_Benefits = 'Yes';
ELSE Eligible_For_Benefits = 'No';
```

Both conditions (Age >= 18 and Employed = 1) must be true for Eligible_For_Benefits to be 'Yes'.

```
IF Age < 18 OR Employed = 0 THEN Eligible_For_Benefits = 'No';
```

In this case, if either condition is true, Eligible_For_Benefits will be 'No'.

7 6. IF with Missing Values

To check for missing values in SAS:

- For numeric variables:

  ```
  IF Age = . THEN Status = 'Missing Age';
  ```

- For character variables:

  ```
  IF Name = '' THEN Status = 'Missing Name';
  ```

8 7. SELECT-WHEN Alternative to IF-THEN-ELSE

For cases where there are many conditions to check, the SELECT-WHEN statement can be clearer and more efficient:

```
SELECT;
    WHEN (Age >= 65) Status = 'Senior';
    WHEN (Age >= 18) Status = 'Adult';
    OTHERWISE Status = 'Minor';
END;
```

This is an alternative to multiple IF-THEN-ELSE statements and can be more readable.

9 8. Nested IF-THEN-ELSE

You can also nest IF-THEN-ELSE blocks for more complex conditions:

```
IF Age >= 18 THEN DO;
    IF Employed = 1 THEN Benefits = 'Yes';
    ELSE Benefits = 'No';
END;
ELSE Benefits = 'N/A';
```

If the person is 18 or older, the inner IF checks their employment status. If the person is younger than 18, Benefits is 'N/A'.

10 Example: Complete SAS Program

```
DATA patients;
    SET healthcare_data;

    IF LOS > 30 THEN Stay_Category = 'Long';
    ELSE IF LOS > 10 THEN Stay_Category = 'Medium';
    ELSE Stay_Category = 'Short';

    IF PSYPROB = -9 THEN DELETE; /* Exclude missing psychiatric problem data */

    IF IDU = 1 AND PSYPROB = 1 THEN IDU_MI = 1;
    ELSE IDU_MI = 0;
RUN;
```

In this example:

- The length of stay (`LOS`) is categorized as `Long`, `Medium`, or `Short`.
- Patients with missing psychiatric problem data (`PSYPROB = -9`) are removed from the dataset.
- A new variable `IDU_MI` is created to identify patients who are both injection drug users and have mental issues.

11 Summary of Logical Operators

- `AND`: Both conditions must be true.
- `OR`: At least one condition must be true.
- `NOT`: Negates a condition.
- `=`: Equal to.
- `^=, NE`: Not equal to.
- `>`: Greater than.
- `>=`: Greater than or equal to.
- `<`: Less than.
- `<=`: Less than or equal to.

Exercises

- Import data from the `samhsa_obs` SAS file.
- Create a new variable with categories for LOS (Length of Stay) as follows:
  - "Long": LOS > 30
  - "Med": $10 <$ LOS ≤ 30
  - "Short": LOS ≤ 10
- Clean the data by creating a new dataset that excludes observations where PSYPROB (Psychiatric Problem) data is missing (missing data is coded as -9).
- For the cleaned dataset, add one variable (e.g., `IDU_MI`) to identify patients who are both Injection Drug Users (IDU = 1) and have mental issues (PSYPROB = 1), such that the new variable value will be 1, and 0 for the rest of the individuals.