

**USING THE IF STATEMENT TO FILTER DATA**

Data File: EXP.DAT

A	1	10	15
A	2	9	12
B	3	11	16
B	4	10	11

**EXAMPLE 1**

**Subsetting IF with Retention**

**SAS Code:**

```
DATA PARTIAL;
    INFILE 'A:\EXP.DAT';
    INPUT SOIL $ TRT COUNT1 COUNT2;

    IF SOIL = 'A';
```

**Result:**

Dataset Partial

A	1	10	15
A	2	9	12

))

**Subsetting IF with Deletion**

**SAS Code:**

```
DATA PARTIAL;
    INFILE 'A:\EXP.DAT';
    INPUT SOIL $ TRT COUNT1 COUNT2;

    IF SOIL = 'A' THEN DELETE;
```

**Result:**

**Dataset Partial**

B	3	11	16
B	4	10	11

**USING THE IF STATEMENT TO FILTER DATA (cont.)**

Sometimes multiple **IF** statements can be speeded up by using the **ELSE** clause.

This is only applicable in certain conditions when one **IF** statement precludes the others.

**EXAMPLE 2**

**SAS Code:**

```

DATA EXP1;
  INFILE 'A:\EXP.DAT';
  INPUT SOIL $ TRT COUNT1 COUNT2;

  IF TRT = 1 THEN NEW = COUNT1 + 5;
  IF TRT = 2 THEN NEW = COUNT1 + 10;
  IF TRT = 3 THEN NEW = COUNT1 + 15;
  IF TRT = 4 THEN NEW = COUNT1 + 20;

))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))

```

```

DATA EXP1;
  INFILE 'A:\EXP.DAT';
  INPUT SOIL $ TRT COUNT1 COUNT2;

  IF TRT = 1 THEN NEW = COUNT1 + 5;
  ELSE IF TRT = 2 THEN NEW = COUNT1 + 10;
  ELSE IF TRT = 3 THEN NEW = COUNT1 + 15;
  ELSE IF TRT = 4 THEN NEW = COUNT1 + 20;

```

## USING THE IF STATEMENT TO FILTER DATA (cont.)

The **ELSE** clause is not always appropriate however.

### EXAMPLE 3

#### SAS Code:

```
DATA EXP1;  
  INFILE 'A:\EXP.DAT';  
  INPUT SOIL $ TRT COUNT1 COUNT2;  
  
  IF TRT = 1 THEN NEW = COUNT1 + 5;  
  
  IF SOIL = 'B' THEN TOTAL = COUNT1 + COUNT2;  
  
  IF SOIL = 'A' AND TRT = 3 THEN DELETE;  
  
  IF COUNT1 <= 10 THEN COUNT1 = COUNT1 + .5;
```

The use of **ELSE** in this case would not give the desired results because when one **IF** is true it does not preclude the possibility of the others.