Frequency Distributions

Absolute and relative frequencies

Suppose there are 10 persons who have participated in a test and their results were declared in two categories as Pass (P) and Fail (F).

P, F, P, F, F, P, P, F, P, P.

Use a_1 and a_2 to refer to Pass and Fail categories.

There are 6 persons who passed, denoted as n_1 = 6.

There are 4 persons who failed, denoted as n_2 = 4.

The number of observations in a particular category is called the **absolute frequency.**

Absolute and relative frequencies

The relative frequency of a_1 is

$$f_1 = \frac{n_1}{n_1 + n_2} = \frac{6}{10} = 0.6 = 60\%$$

The relative frequency of a_2 is

$$f_2 = \frac{n_2}{n_1 + n_2} = \frac{4}{10} = 0.4 = 40\%$$

This gives us information about the proportions of Pass and Fail persons in the test.

Absolute and relative frequencies

`table(data vector)` creates the absolute frequency of the `data vector` of the given data in the vector.

Enter data as `x`

```
table(x) # absolute frequencies
```

```
table(x)/length(x) # relative frequencies
```

Absolute and relative frequencies

Results of 10 persons declared in two categories as Pass (P) and Fail (F) is categorised as 1 and 2 respectively.

P, F, P, F, F, P, P, F, P, P

1, 2, 1, 2, 2, 1, 1, 2, 1, 1

```
> result <- c(1, 2, 1, 2, 2, 1, 1, 2, 1, 1)
> result
 [1] 1 2 1 2 2 1 1 2 1 1
```

Absolute and relative frequencies

```
> table(result) # Absolute frequencies
result
1 2
6 4

> table(result)/length(result) #Relative freq.
result
  1   2
0.6 0.4
```

Frequency Distribution

- **Arrangement of ungrouped data in the form of group is called frequency distribution of data.**

- **Classify the data into different classes by dividing the entire range of the values of variables into suitable number of groups called class.**

Frequency Distribution

- Lower and upper boundary figures of a class are called the lower limit and upper limit respectively.
- Difference between the limits is called the width of the class or class interval.
- The value of variate lies in the middle of lower and upper limits.

Frequency Distribution

- The number of observations in a particular class is called absolute frequency or frequency.
- The number of observations in a particular class divided by total frequency is called relative frequency.
- The cumulative frequency corresponding to any variate value is the number of observations less than or equal to that value.
- The cumulative frequency corresponding to a class is the total number of observations less than or equal to the upper limit of the class.

Frequency Distribution - Example

Following are the time taken (in seconds) by 20 participants in a race:

32, 35, 45, 83, 74, 55, 68, 38, 35, 55, 66, 65, 42, 68, 72, 84, 67, 36, 42, 58.

```
> time <- c(32, 35, 45, 83, 74, 55, 68, 38, 35, 55,
66, 65, 42, 68, 72, 84, 67, 36, 42, 58)
> time

 [1] 32 35 45 83 74 55 68 38 35 55 66 65 42 68

72 84 67 36 42 58
```

First step is to find the range of the data values which can be partitioned into class interval.

Use command `range` which returns a vector containing the minimum and maximum of all the given arguments.

Usage:

`range(data vector)` returns a vector containing the minimum and maximum of all the given arguments.

Frequency Distribution

Example (contd.):

```
> range(time)
[1] 32 84
```

This result gives an information and it looks reasonable to divide the data in class following intervals:

31-40, 41-50, 51-60, 61-70, 71-80 and 81-90

Create a sequence starting from 30 to 90 at an interval of 10 integers denoting the width.

Frequency Distribution

Example (contd.):

Create a sequence starting from 30 to 90 at an interval of 10 integers denoting the width.

```
> breaks = seq(30, 90, by=10)   # sequence at
                                  interval of 10 integers
> breaks
[1] 30 40 50 60 70 80 90
```

Frequency Distribution

Class intervals	Mid point	Absolute frequency (or frequency)	Relative Frequency	Cumulative Frequency
31 – 40	35.5	5	5/20 = 0.25	5
41 – 50	45.5	3	3/20 = 0.15	5+3 = 8
51 – 60	55.5	3	3/20 = 0.15	5+3+3 = 11
61 – 70	65.5	5	5/20 = 0.25	5+3+3+5 = 16
71 – 80	75.5	2	2/20 = 0.01	5+3+3+5+2 = 18
81 - 90	85.5	2	2/20 = 0.01	5+3+3+5+2+2 = 20
	Total	20	1	

Frequency Distribution

Now we need to convert Numeric to Factor using a command `cut`

Usage: `cut(data vector, breaks, right = FALSE)` divides the range of `data vector` into intervals and codes the values in `data vector` according to which interval they fall.

`breaks` is a numeric vector of two or more unique cut points or a single number (greater than or equal to 2) giving the number of intervals into which `data vector` is to be cut.

As the intervals are to be closed on the left, and open on the right, we set the right argument as `FALSE`.

Frequency Distribution

Now we classify the time data according to the width intervals with

`cut`.

```
> time.cut = cut(time,breaks,right=FALSE)
> time.cut
 [1] [30,40) [30,40) [40,50) [80,90) [70,80) [50,60) [60,70)
 [8] [30,40) [30,40) [50,60) [60,70) [60,70) [40,50) [60,70)
[15] [70,80) [80,90) [60,70) [30,40) [40,50) [50,60)
Levels: [30,40) [40,50) [50,60) [60,70) [70,80) [80,90)
```

Frequency Distribution

Interpretation of outcome

```
> time
[1] 32 35 45 83 74 55 68 38 35 55 66 65 42 68 72 84 67 36 42 58

> time.cut
 [1] [30,40) [30,40) [40,50) [80,90) [70,80) [50,60) [60,70)
 [8] [30,40) [30,40) [50,60) [60,70) [60,70) [40,50) [60,70)
[15] [70,80) [80,90) [60,70) [30,40) [40,50) [50,60)
Levels: [30,40) [40,50) [50,60) [60,70) [70,80) [80,90)
```

Frequency Distribution

Now we can compute the absolute frequency of time data in each width interval with the `table` function

`table(variable)` creates the absolute frequency of the `variable` of the data file which generates the frequency distribution of the data on `variable`.

Frequency Distribution

```
> table(time.cut)
time.cut
[30,40) [40,50) [50,60) [60,70) [70,80) [80,90)
      5       3       3       5       2       2
```

Use the `cbind` function to print the frequency distribution in column format.

```
> cbind(table(time.cut))
        [,1]
[30,40)    5
[40,50)    3
[50,60)    3
[60,70)    5
[70,80)    2
[80,90)    2
```

Frequency Distribution

To compute the relative frequency of time data in each width interval with the `table` function with `length` function

`table(variable)/length(variable)` creates the relative frequency of the `variable` of the data file which generates the frequency distribution of the data on `variable`.

Frequency Distribution

Example (contd.):

```
> table(time.cut)/length(time.cut)
time.cut
[30,40) [40,50) [50,60) [60,70) [70,80) [80,90)
   0.25    0.15    0.15    0.25    0.10    0.10
```

Use the **cbind** function to print the frequency distribution in column format.

```
> cbind(table(time.cut)/length(time.cut))
        [,1]
[30,40) 0.25
[40,50) 0.15
[50,60) 0.15
[60,70) 0.25
[70,80) 0.10
[80,90) 0.10
```

Cumulative Distribution Function (CDF) for data

It gives us an idea about the cumulative frequencies up to a certain point.

The cumulative frequencies are computed by the function `cumsum`

Usage: `cumsum(table(variable))` returns a vector whose elements are the cumulative sums of the elements of the frequencies in the `variable` in the argument.

Cumulative Distribution Function (CDF) for data

Example (contd.):

```
> cumsum(table(time.cut))

[30,40) [40,50) [50,60) [60,70) [70,80) [80,90)
      5       8      11      16      18      20
```

Use the `cbind` function to print the cumulative frequency distribution in column format.

```
> cbind(cumsum(table(time.cut)))

        [,1]
[30,40)    5
[40,50)    8
[50,60)   11
[60,70)   16
[70,80)   18
[80,90)   20
```

Cumulative Distribution Function (CDF) for data

If the cumulative frequencies are to be computed based on <u>relative frequency</u> then the function `cumsum` is used with

`table(variable)/length(variable)`

Usage: `cumsum(table(variable)/length(variable))`

returns a vector whose elements are the cumulative sums of the elements of the relative frequencies in the `variable` in the argument.

Cumulative Distribution Function (CDF) for data

```
> cumsum(table(time.cut)/length(time.cut))
[30,40) [40,50) [50,60) [60,70) [70,80) [80,90)
   0.25    0.40    0.55    0.80    0.90    1.00

> cbind(cumsum(table(time.cut)/length(time.cut)))
        [,1]
[30,40) 0.25
[40,50) 0.40
[50,60) 0.55
[60,70) 0.80
[70,80) 0.90
[80,90) 1.00
```