Descriptive Statistics With R Software

Variation in Data

::

Absolute Deviation and Absolute Mean Deviation

Shalabh

Department of Mathematics and Statistics

Indian Institute of Technology Kanpur

Deviation Based Measures of Variation

Range, interquartile range and quartile deviation are based on specific values and partitioning values.

Need a measure which can measure the deviation of every observation around any given value.

Deviation Based Measures of Variation

Deviation of any observation x_i from any value A is $d_i = (x_i - A)$.

If $x_i > A$, then such deviations d_i's are positive.

If $x_i < A$, then such deviations d_i's are negative.

If $x_i = A$, then such deviations d_i's are zero.

Deviation Based Measures of Variation

If we consider the average of these deviations *d_i's*, then the average value $\frac{1}{n}\sum_{i=1}^{n} d_i$ may be close to zero and will be reflecting that there is no variation or small variation, which may not be correct.

So we need not to consider the signs of the deviations.

We need to consider ONLY the magnitudes of the deviations.

Deviation Based Measures of Variation

Two options:

1. **Consider absolute values of these deviations.**

2. **Consider squared values of these deviations.**

Notations for Ungrouped (Discrete) Data

Observations on a variable *X* are obtained as $x_1, x_2, \ldots, x_n$.

Notations for Grouped (Continuous) data

Observations on a variable X are obtained and tabulated in K class intervals in a frequency table as follows. The mid points of the intervals are denoted by $x_1, x_2, \ldots, x_k$ which occur with frequencies $f_1, f_2, \ldots, f_K$ respectively and $n = f_1 + f_2 + \ldots + f_K$.

Class intervals	Mid point (m_i)	Absolute frequency (f_i)
$e_1 - e_2$	$x_1 = (e_1 + e_2)/2$	f_1
$e_2 - e_3$	$x_2 = (e_2 + e_3)/2$	f_2
…	…	…
$e_{K-1} - e_K$	$x_K = (e_{K-1} + e_K)/2$	f_K

Absolute Deviation

The absolute deviation of observations $x_1, x_2, \ldots, x_n$ around a value A

is defined as

- $D(A) = \frac{1}{n}\sum_{i=1}^{n}|x_i - A|$ **for discrete (ungrouped) data.**

- $D(A) = \frac{1}{n}\sum_{i=1}^{K} f_i |x_i - A|$ **for continuous (grouped) data.**

  where $n = \sum_{i=1}^{K} f_i$

Absolute Mean Deviation

The absolute deviation of observations $x_1, x_2, \ldots, x_n$ is minimum when measured around median, i.e., *A* is the median of data.

In this case, the absolute deviation is termed as absolute mean deviation and is defined as

- $D(\bar{x}_{med}) = \frac{1}{n}\sum_{i=1}^{n} \left| x_i - \bar{x}_{med} \right|$ **for discrete (ungrouped) data.**

- $D(\bar{x}_{med}) = \frac{1}{n}\sum_{i=1}^{K} f_i \left| x_i - \bar{x}_{med} \right|$ **for continuous (grouped) data.**

  where $n = \sum_{i=1}^{K} f_i$

Absolute Mean Deviation

The absolute mean deviation measures the spread and scatterdness of data around, preferebly the median value, in terms of absolute deviations.

Absolute Deviation and Absolute Mean Deviation
Decision Making

The data set having higher value of absolute mean deviation (or absolute deviation) has more variability.

The data set with lower value of absolute mean deviation (or absolute deviation) is preferable.

If we have two data sets and suppose their absolute mean deviation are AMD_1 and AMD_2.

If $AMD_1 > AMD_2$ then the data in AMD_1 is said to have more variability than the data in AMD_2.

Absolute Deviation and Absolute Mean Deviation

R command: Ungrouped data

Data vector: `x`

Absolute deviation for given *A*

```
mean(abs(x - A))
```

Absolute mean deviation

```
mean(abs(x - median(x)))
```

Absolute Deviation and Absolute Mean Deviation

R command: Ungrouped data and missing values

If data vector `x` has missing values as `NA`, say `xna`, then R command is

Absolute deviation for given *A*

```
mean(abs((xna - A)),na.rm=TRUE)
```

Absolute mean deviation

```
mean(abs((xna - median(xna, na.rm=TRUE))),
na.rm= TRUE)
```

Absolute Deviation and Absolute Mean Deviation

R command: Grouped data

Data vector: `x`

Frequency vector: `f`

Absolute deviation for given *A*

```
sum(f * abs(x - A))/sum(f)
```

Absolute mean deviation

```
sum(f * abs(x - xmedian))//sum(f)
```

Note: Median in this case is to be computed as `xmedian` using the median for grouped data separately.

Absolute Deviation and Absolute Mean Deviation

Example: Ungrouped data

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)

> A = 10
> mean(abs((time - A)))  #Absolute deviation around A= 10
[1] 46

> median(time)
[1] 56.5

> mean(abs(time - median(time)))  # Absolute mean deviation around median
 [1] 14.5
```

Absolute Deviation and Absolute Mean Deviation

Example: Ungrouped data

R Console

```
> time
 [1] 32 35 45 83 74 55 68 38 35 55 66 65 42 68 72 84 67 36 42 58
> A=10
> A
[1] 10
> mean(abs(time - A))
[1] 46
>
> median(time)
[1] 56.5
> mean(abs(time - mean(time)))
[1] 14.5
> 
```

Absolute Deviation and Absolute Mean Deviation

Example: Grouped data

Considering the data as grouped data, we can present the data as

Class intervals	Mid point	Absolute frequency (or frequency)
31 – 40	35.5	5
41 – 50	45.5	3
51 – 60	55.5	3
61 – 70	65.5	5
71 – 80	75.5	2
81 - 90	85.5	2
	Total	20

We need to find the frequency vector and median.

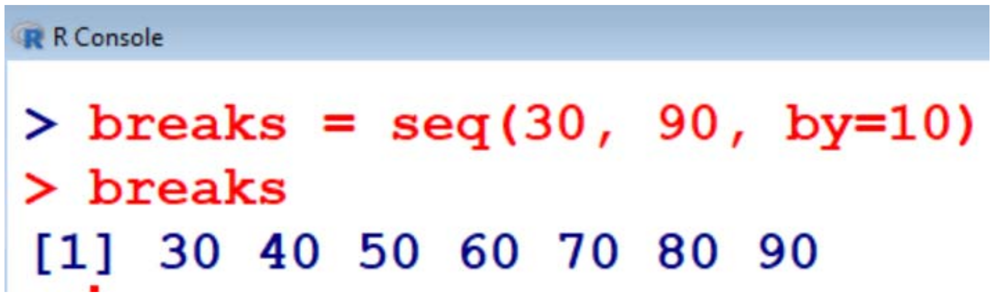
Absolute Deviation and Absolute Mean Deviation

Example: Grouped data - Obtaining frequencies:

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

```
breaks = seq(30, 90, by=10)    # Sequence of 10 integers
                                  at interval of 10
```

```
> breaks = seq(30, 90, by=10)
> breaks
[1] 30 40 50 60 70 80 90
```

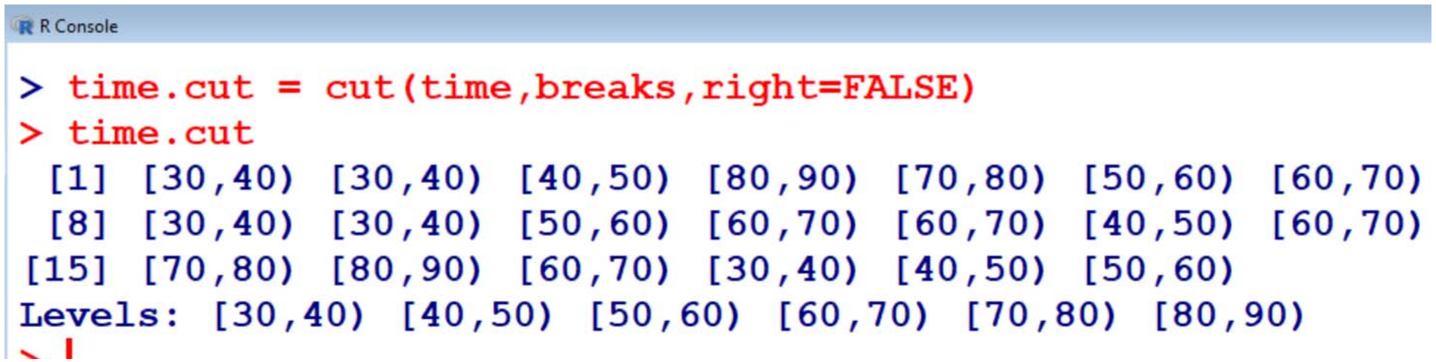


Absolute Deviation and Absolute Mean Deviation

Example: Grouped data - Obtaining frequencies:

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)
```


R Console
> 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)

Absolute Deviation and Absolute Mean Deviation

Example: Grouped data - **Obtaining frequencies:**

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
```

Extract frequencies from frequency table using command

```
> f = as.numeric(table(time.cut))
> f
[1] 5 3 3 5 2 2
```

Absolute Deviation and Absolute Mean Deviation

Example: Grouped data - Obtaining mid points:

Mid points, as obtained from the frequency table, are

```
> x = c(35,45,55,65,75,85)

> x
[1] 35 45 55 65 75 85
```

Note that the mid points are obtained from the frequency table obtained from the R software

```
[30,40) [40,50) [50,60) [60,70) [70,80) [80,90)
```

Absolute Deviation and Absolute Mean Deviation

Example: Grouped data – Obtaining median

Obtain median from the frequency table using

Median class (m = 3) : 50 – 60

Lower limit of class (e_m) = e_3 = 50

Frequency of median class (f_m) = f_3 = 3/20

Width of median class (d_m) = d_3 = 50 – 60 = 10

$$\bar{x}_{med} = e_m + \frac{0.5 - \sum_{j=1}^{m-1} f_i}{f_m} d_m$$

$$= 50 + \frac{0.5 - \left(\frac{5}{20} + \frac{3}{20}\right)}{3/20} \times 10$$

$$\approx 56.66$$

Absolute Deviation and Absolute Mean Deviation

Example: Grouped data

```
> f = c(5,3,3,5,2,2)

> x = c(35,45,55,65,75,85)

> xmedian = 56.66

> A = 10

> sum(f * abs(x - A))/sum(f)  #Absolute deviation
[1] 46                                    around A= 10

> sum(f * abs(x - xmedian))/sum(f)  # Absolute
[1] 14.166                 mean deviation around median
```

Absolute Deviation and Absolute Mean Deviation

Comparison of results:

Ungrouped data

```
> mean(abs((time - A)))
[1] 46
```
 # Absolute deviation around *A*= 10

```
> mean(abs(time - median(time)))
[1] 14.5
```
 # Absolute mean

Grouped data

```
> sum(f * abs(x - A))/sum(f)
[1] 46
```
 # Absolute deviation around *A*= 10

```
> sum(f * abs(x - xmedian))/sum(f)
[1] 14.166
```
 # Absolute mean deviation around median

Absolute Deviation and Absolute Mean Deviation

Example: **Handling missing values**

Suppose two data points are missing in the earlier example where the time taken (in seconds) by 20 participants in a race. They are recorded as NA

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

```
> time.na = c(NA, NA, 45, 83, 74, 55, 68, 38,
35, 55, 66, 65, 42, 68, 72, 84, 67, 36, 42, 58)
```

Absolute Deviation and Absolute Mean Deviation

Example: Handling missing values

```
> time.na = c(NA, NA, 45, 83, 74, 55, 68, 38,
35, 55, 66, 65, 42, 68, 72, 84, 67, 36, 42, 58)

> A = 10

> mean(abs((time.na - A)), na.rm= TRUE)

[1] 48.5


> median(time.na, na.rm = TRUE)
[1] 61.5

> mean(abs((time.na - median(time.na, na.rm =
TRUE))), na.rm= TRUE)
[1] 13.38889
```

Absolute Deviation and Absolute Mean Deviation

Example: Handling missing values

R Console

```
> time.na
 [1] NA NA 45 83 74 55 68 38 35 55 66 65 42 68 72 84 67 36 42 58
> A =10
> A
[1] 10
> mean(abs((time.na - A)), na.rm= TRUE) #Absolute deviation around A= 10
[1] 48.5
>
> median(time.na, na.rm=TRUE)
[1] 61.5
> mean(abs((time.na - median(time.na, na.rm = TRUE))), na.rm= TRUE) # Absolute mean d$
[1] 13.38889
> |
```