# VectorAnalysis_Rpackage

1) Package for Vector Analysis contains 7 operations:
    - smallest:
      Get minmum element in vector. Input to this function is a vector.

    - largest:
      Get maximum element in vector. Input to this function is a vector.

    - get_outliers
      Return outliers in vector. Input to this function is a vector.

    - remove_outliers
      Return input vector but without outliers. Input to this function is a vector.

    - min_max_normalize
      Normalize input vector according to min max interval.
      The input is two vector.First vector is your data that you want to normalize.
      The second vector contains two numbers which are your normalization interval.

    - z_score_normalize:
      Normalize input vector accoring to z score.

    - decimal_normalize:
      Normalize input vector accoring to decimal method.

2) Examples:

- smallest and largest:

```
> library(vectorAnalysis)
> Dataset=c(150000, 320000, 2500000, 180000, 120000, 85000)
> library(vectorAnalysis)
> Dataset=c(150000, 320000, 2500000, 180000, 120000, 85000)
> smallest(Dataset)
[1] 85000
> largest(Dataset)
[1] 2500000
```

- get_outliers:
  Theoritical Background Example:

| Car Model | Manufacturing Year | Cylinders | CC | Price (L.E) | Maximum Speed (km/hr) | Horse Power | Used/New | Usage Duration (Years) |
|---|---|---|---|---|---|---|---|---|
| Toyota | 2009 | | 1600 | 150000 | 195 | 1400 | New | 3 |
| Jeep | 2009 | 6 | 3700 | 320000 | 200 | 210 | New | 0 |
| Mercury | MWVR09 | 6 | 4000 | 2500000 | | 210 | | 0 |
| Opel | 2008 | 4 | 1600 | 180000 | 192 | 105 | New | 10 |
| Mitsubishi | 2006 | 4 | -1600 | | 170 | 106 | Used | 3 |
| Honda | 2009 | 4 | 1500 | 120000 | 180 | 92 | New | 0 |
| Mazda | 2003 | 4 | 1600 | 85000 | 180 | | Used | 6 |

Find any outliers in field "Price."
Dataset = 150000, 320000, 2500000, 180000, 120000, 85000
Ordered Dataset = 85000, 120000, 150000, 180000, 320000, 2500000
Q2 = (150000 + 180000) / 2 = 165000
Q1 = 120000
Q3 = 320000
IQR = Q3 – Q1 = 200000
Q1 – (1.5)*(IQR) = -180000
Q3 + (1.5)*(IQR) = 620000
As (2500000) > (620000) Hence ➡ 2500000 is an outfiller

Using the Package:

```
> library(vectorAnalysis)
> Dataset=c(150000, 320000, 2500000, 180000, 120000, 85000)
> get_outliers(Dataset)
[1] 2500000
```

- remove_outliers:

```
> library(vectorAnalysis)
> Dataset=c(150000, 320000, 2500000, 180000, 120000, 85000)
> remove_outliers(Dataset)
[1] 150000 320000 180000 120000  85000
```

- min_max_normalize:
  Theoritical Background Example:

| Planet | Average Surface Temperature (°C) | Approximate Solar Day (Earth Days) | Approximate Solar Year (Earth Days) | Approximate Diameter (km) | Average Distance to Sun (km) |
|---|---|---|---|---|---|
| Mercury | 179 | 58.65 | 88 | 4880 | 58000000 |
| Venus | 482 | 243 | 225 | 12102 | 108000000 |
| Mars | -60 | 1 | 687 | 6792 | 230000000 |
| Saturn | -153 | 0.4263 | 10760 | 120000 | 1427000000 |
| Uranus | -218 | 0.7458 | 30681 | 50800 | 2870000000 |

$$v' = \left(\left[\frac{v - min_A}{max_A - min_A}\right] \times [new\_max_A - new\_min_A]\right) + new\_min_A$$

- Approximate Solar Day:
  $max_A = 243$ $min_A = 0.4263$ $new\_max_A = 1$ $new\_min_A = 0$
- Approximate Solar Year:
  $max_A = 30681$ $min_A = 88$ $new\_max_A = 1$ $new\_min_A = 0$
- Approximate Diameter:
  $max_A = 120000$ $min_A = 4880$ $new\_max_A = 1$ $new\_min_A = 0$

| Approximate Solar Day (Earth Days) | Approximate Solar Year (Earth Days) | Approximate Diameter (km) |
|---|---|---|
| 0.24 | 0 | 0 |
| 1 | 0.0045 | 0.063 |
| 0.0024 | 0.020 | 0.017 |
| 0 | 0.35 | 1 |
| 0.0013 | 1 | 0.40 |

Using the Package:

```
> library(vectorAnalysis)
> Dataset=c(88,225,687,10760,30681)
> newInterval=c(0,1)
> min_max_normalize(Dataset,newInterval)
[1] 0.000000000 0.004478149 0.019579642 0.348837969 1.000000000
```

- z_score_normalize:
  Theoritical Background Example ➡ [From the Previous Table in the example]

$$v' = \frac{v - \bar{A}}{\sigma_A} \qquad \bar{A} = \frac{\sum_{i=1}^{n} A}{n} \qquad \sigma_A = \sqrt{\frac{\sum_{i=1}^{n} (A_i - \bar{A})^2}{n-1}}$$

  - Approximate Solar Day:
    $\bar{A} = 60.76$ $\sigma_A = 104.92$
  - Approximate Solar Year:
    $\bar{A} = 8488.2$ $\sigma_A = 13204.01$
  - Approximate Diameter:
    $\bar{A} = 38914.8$ $\sigma_A = 49054.14$

| Approximate Solar Day (Earth Days) | Approximate Solar Year (Earth Days) | Approximate Diameter (km) |
|---|---|---|
| -0.020 | -0.64 | -0.69 |
| 1.74 | -0.63 | -0.55 |
| -0.57 | -0.59 | -0.65 |
| -0.58 | 0.17 | 1.65 |
| -0.57 | 1.68 | 0.24 |

Using the Package:

```
> library(vectorAnalysis)
> Dataset=c(88,225,687,10760,30681)
> z_score_normalize(Dataset)
[1] -0.6361853 -0.6258097 -0.5908203  0.1720537  1.6807616
>
```

- decimal_normalize:
  Theoritical Background Example ➡ [From the Previous Table in the example]
  $v' = v / 10^j$
  - Approximate Solar Day: $j = 3$
  - Approximate Solar Year: $j = 5$
  - Approximate Diameter: $j = 6$

| Approximate Solar Day (Earth Days) | Approximate Solar Year (Earth Days) | Approximate Diameter (km) |
|---|---|---|
| 0.059 | 0.00088 | 0.00488 |
| 0.243 | 0.00225 | 0.0121 |
| 0.001 | 0.00687 | 0.00679 |
| 0.00042 | 0.1076 | 0.12 |
| 0.0007 | 0.30681 | 0.051 |

Using the Package:

```
> Dataset=c(88,225,687,10760,30681)
> decimal_normalize(Dataset)
[1] 0.00088 0.00225 0.00687 0.10760 0.30681
>
```