

# Package: TransP (via r-universe)

October 29, 2024

**Type** Package

**Title** Implementation of Transportation Problem Algorithm

**Version** 0.1

**Date** 2016-02-11

**Author** Somenath Sit

**Maintainer** Somenath Sit <somenath.bhu.2010@gmail.com>

**Description** Implementation of two transportation problem algorithms.

1. North West Corner Method 2. Minimum Cost Method or Least cost method. For more technical details about the algorithms please refer below URLs.

<<http://www.universalteacherpublications.com/univ/ebooks/or/Ch5/nw.htm>>.

<<http://personal.maths.surrey.ac.uk/st/J.F/chapter7.pdf>>.

**License** GPL (>= 2)

**LazyLoad** yes

**RoxygenNote** 5.0.1

**Repository** <https://somenath24.r-universe.dev>

**RemoteUrl** <https://github.com/somenath24/transp>

**RemoteRef** HEAD

**RemoteSha** 35fbde71628c9d898c56619ed4d90ed4d447851f

## Contents

mincost . . . . .	2
nwc . . . . .	3
<b>Index</b>	<b>5</b>

mincost

*Implements Minimum Cost Algorithm to solve transportation problem***Description**

This function implements Minimum Cost Algorithm to resolve transportation problem and get optimized allocation matrix

**Usage**

```
mincost(ex_matrix)
```

**Arguments**

**ex\_matrix** A cost matrix where last column must be the supply and last row must be the demand. Input matrix should not have any missing values (NA), otherwise function will throw an error.

**Details**

This function takes a cost matrix (with Supply and Demand) and using North-West Corner approach gives the allocation matrix as well as the calculated optimized cost. This function checks for degenerated problem but it can't resolve it. User need to resolve by seeing final allocation matrix. If Supply and Demand are not equal Balance Supply/Demand will be stored in Dummy variable.

**Value**

A List which contains the allocation matrix and the total optimized cost.

**Examples**

```
## Not run:
```

```
#Input matrix where last row is the Demand and last column is the Supply
ex_matrix=data.frame(M1=c(13,10,25,17,210),M2=c(25,19,10,24,240),
                     M3=c(8,18,15,18,110),M4=c(13,5,14,13,80),M5=c(20,12,18,19,170),
                     Supply=c(430,150,100,130,810),
                     row.names = c("W1","W2","W3","W4","Demand"))
```

```
ex_matrix
      M1 M2 M3 M4 M5 Supply
W1     13 25  8 13 20   430
W2     10 19 18  5 12   150
W3     25 10 15 14 18   100
W4     17 24 18 13 19   130
Demand 210 240 110 80 170   810
```

```
mincost(ex_matrix)
```

```

$Alloc_Matrix
      M1  M2  M3  M4  M5
W1 140 140 110  0  40
W2  70  0   0  80  0
W3  0 100  0  0  0
W4  0  0  0  0 130

$Total_Cost
[1] 11570

## End(Not run)

```

nwc

---

*Implements North-West Corner Algorithm to solve transportation problem*

---

## Description

This function implements North-West Corner Algorithm to solve transportation problem by optimized cost matrix and total optimized cost

## Usage

```

# Get optimized cost matrix for input matrix ex_matrix
nwc(ex_matrix)

```

## Arguments

ex_matrix	A cost matrix where last column must be the supply and last row must be the demand. Input matrix should not have any missing values (NA), otherwise function will throw an error.
-----------	---

## Details

This function takes a cost matrix (with Supply and Demand) and using North-West Corner approach gives the cost allocation matrix as well as the calculated optimized cost. This function checks for degenerated problem but it can't resolve it. User need to resolve by seeing the cost allocation matrix.

## Value

A List which contains the Cost allocation matrix and the total optimized cost.

## Examples

## Not run:

```
#Input matrix where last row is the Demand and last column is the Supply
ex_matrix=data.frame(M1=c(13,10,25,17,210),M2=c(25,19,10,24,240),
                     M3=c(8,18,15,18,110),M4=c(13,5,14,13,80),M5=c(20,12,18,19,170),
                     Supply=c(430,150,100,130,810),
                     row.names = c("W1","W2","W3","W4","Demand"))
```

```
ex_matrix
      M1 M2 M3 M4 M5 Supply
W1     13 25  8 13 20   430
W2     10 19 18  5 12   150
W3     25 10 15 14 18   100
W4     17 24 18 13 19   130
Demand 210 240 110 80 170   810
```

```
nwc(ex_matrix)
$Alloc_Matrix
      M1 M2 M3 M4 M5
W1 210 220  0  0  0
W2  0 20 110 20  0
W3  0  0  0 60 40
W4  0  0  0  0 130
```

```
$Total_Cost
[1] 14720
```

## End(Not run)

# Index

mincost, [2](#)

nwc, [3](#)