QMM_ASSIGNMENT DEA

Yeswanth Siripurapu

2023-11-06

The code will read the file energy.csv and store all of the data in a

#DataFrame called Energy. #The input code is too short to provide a detailed and accurate answer. To gain deeper insight, try again using a longer piece of code.

```
Energy <- read.csv("D:/energy.csv")
```

#The code is a benchmarking code that will be run in the R environment. #The input code is too short to provide a detailed and accurate answer. To gain deeper insight, try again using a longer piece of code.

```
library(Benchmarking)

## Warning: package 'Benchmarking' was built under R version 4.3.2

## Loading required package: lpSolveAPI

## Loading required package: ucminf

## Loading required package: quadprog
```

#The code is a function that takes in two vectors and returns the correlation #between them. #The first vector is the size of the power grid, which is stored as D.C..Size, #and the second vector is how many times it has shut down over time, which is stored as Shut.Downs. #The code starts by creating an empty list called e with three elements: x, y, #and RTS="crs". #This means that x will be used to store data about the size of power grids #(D.C.), y will be used to store data about how often they have shut down #(Shutdowns), and RTS=3 tells us what type of correlation we are looking for: #Pearson's r-squared value 3 or Spearman's rank order correlation coefficient 3 #The code is designed to create a time series plot of energy consumption and queue time.

```
x1=Energy$D.C..Size
x2=Energy$Shut.Downs
y1=Energy$Computing.Time..h.
x=matrix(c(x1,x2),ncol=2)
y2=Energy$MWh.Consumed
y3=Energy$Queue.Time..ms.
y=matrix(c(y1,y2,y3),ncol=3)
colnames(y)= c("Computing.Time..h.","MWh.Consumed","Queue.Time..ms.")
colnames(x)=c("D.C..Size","Shut.Downs")
x
```

```
##       D.C..Size Shut.Downs
##  [1,]      1000      37166
##  [2,]      1000      13361
##  [3,]      1000      14252
##  [4,]      1000      36404
##  [5,]      1000      19671
##  [6,]      1000      32407
##  [7,]      5000       6981
##  [8,]      5000       9877
##  [9,]      5000      33589
## [10,]      5000       8578
## [11,]      5000      11863
## [12,]      5000      15452
## [13,]     10000       9680
## [14,]     10000      11388
## [15,]     10000      18150
## [16,]     10000      18409
## [17,]     10000      29707
## [18,]     10000      40772
```

```
y
```

```
##       Computing.Time..h. MWh.Consumed Queue.Time..ms.
##  [1,]             104.42        49.01            90.1
##  [2,]             104.26        49.65          1093.0
##  [3,]             104.17        49.60             0.1
##  [4,]              49.25        23.92            78.3
##  [5,]              49.63        24.65          1188.7
##  [6,]              49.34        24.19             1.1
##  [7,]              99.96       237.09           126.2
##  [8,]              99.96       235.92           129.8
##  [9,]             100.03       234.90          1122.6
## [10,]             100.26       239.13             0.7
## [11,]             100.26       236.95             1.0
## [12,]              46.70       115.82             0.5
## [13,]             101.56       481.36           325.2
## [14,]             101.56       479.36           327.9
## [15,]             101.63       486.11             2.6
## [16,]             101.63       484.69             2.5
## [17,]              45.83       228.31          1107.6
## [18,]              46.09       233.50             3.8
```

```
e=dea(x,y,RTS="crs")
e
```

```
##  [1] 1.0000 1.0000 0.9991 0.4818 1.0000 0.4872 1.0000 0.9826 0.9578 1.0000
## [11] 0.9806 0.4754 1.0000 0.9944 1.0000 0.9970 0.5290 0.4783
```

```
peers(e)
```

```
##       peer1 peer2 peer3
##  [1,]     1    NA    NA
##  [2,]     2    NA    NA
##  [3,]     1     2    NA
##  [4,]     2    NA    NA
##  [5,]     5    NA    NA
##  [6,]     2    NA    NA
##  [7,]     7    NA    NA
##  [8,]     2    10    13
##  [9,]     2    15    NA
## [10,]    10    NA    NA
## [11,]     2    13    15
## [12,]     2    15    NA
## [13,]    13    NA    NA
## [14,]     2    13    15
## [15,]    15    NA    NA
## [16,]     2    15    NA
## [17,]     2    13    NA
## [18,]     2    15    NA
```

```
lambda(e)
```

```
##                L1         L2 L5 L7       L10       L13        L15
##  [1,] 1.000000000 0.00000000  0  0 0.0000000 0.0000000 0.00000000
##  [2,] 0.000000000 1.00000000  0  0 0.0000000 0.0000000 0.00000000
##  [3,] 0.009970484 0.98915099  0  0 0.0000000 0.0000000 0.00000000
##  [4,] 0.000000000 0.48177241  0  0 0.0000000 0.0000000 0.00000000
##  [5,] 0.000000000 0.00000000  1  0 0.0000000 0.0000000 0.00000000
##  [6,] 0.000000000 0.48721047  0  0 0.0000000 0.0000000 0.00000000
##  [7,] 0.000000000 0.00000000  0  1 0.0000000 0.0000000 0.00000000
##  [8,] 0.000000000 0.22098286  0  0 0.5914729 0.1734861 0.00000000
##  [9,] 0.000000000 2.03346741  0  0 0.0000000 0.0000000 0.27553094
## [10,] 0.000000000 0.00000000  0  0 1.0000000 0.0000000 0.00000000
## [11,] 0.000000000 0.53626578  0  0 0.0000000 0.4082527 0.02840485
## [12,] 0.000000000 0.26256674  0  0 0.0000000 0.0000000 0.21144095
## [13,] 0.000000000 0.00000000  0  0 0.0000000 1.0000000 0.00000000
## [14,] 0.000000000 0.04516562  0  0 0.0000000 0.8554257 0.13443418
## [15,] 0.000000000 0.00000000  0  0 0.0000000 0.0000000 1.00000000
## [16,] 0.000000000 0.02236541  0  0 0.0000000 0.0000000 0.99479451
## [17,] 0.000000000 0.89985422  0  0 0.0000000 0.3814863 0.00000000
## [18,] 0.000000000 0.93720988  0  0 0.0000000 0.0000000 0.38461980
```

```
#dea.plot.transform(y[,1],y[,2],RTS=3)
```