Ch10. Reshaping data

Jongrak Jeong

2023-01-19

```
setwd("~/Library/Mobile Documents/com~apple~CloudDocs/Study/2_Data Science/Practice/R Programming by He
```

1. stacking

load the dataset

```
library(datasets)
data(PlantGrowth)
str(PlantGrowth)
```

```
## 'data.frame':    30 obs. of  2 variables:
##  $ weight: num  4.17 5.58 5.18 6.11 4.5 4.61 5.17 4.53 5.33 5.14 ...
##  $ group : Factor w/ 3 levels "ctrl","trt1",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
PlantGrowth
```

```
##    weight group
## 1    4.17  ctrl
## 2    5.58  ctrl
## 3    5.18  ctrl
## 4    6.11  ctrl
## 5    4.50  ctrl
## 6    4.61  ctrl
## 7    5.17  ctrl
## 8    4.53  ctrl
## 9    5.33  ctrl
## 10   5.14  ctrl
## 11   4.81  trt1
## 12   4.17  trt1
## 13   4.41  trt1
## 14   3.59  trt1
## 15   5.87  trt1
## 16   3.83  trt1
## 17   6.03  trt1
## 18   4.89  trt1
## 19   4.32  trt1
## 20   4.69  trt1
## 21   6.31  trt2
## 22   5.12  trt2
## 23   5.54  trt2
## 24   5.50  trt2
## 25   5.37  trt2
## 26   5.29  trt2
## 27   4.92  trt2
## 28   6.15  trt2
## 29   5.80  trt2
## 30   5.26  trt2
```

```
PlantGrowth[sample(1:nrow(PlantGrowth), 10), ]
```

```
##    weight group
## 5    4.50  ctrl
## 10   5.14  ctrl
## 20   4.69  trt1
## 14   3.59  trt1
## 8    4.53  ctrl
## 18   4.89  trt1
## 27   4.92  trt2
## 15   5.87  trt1
## 28   6.15  trt2
## 22   5.12  trt2
```

ANOVA

```
anova(with(PlantGrowth, lm(weight ~ group)))
```

```
## Analysis of Variance Table
##
## Response: weight
##           Df  Sum Sq Mean Sq F value  Pr(>F)
## group      2  3.7663  1.8832  4.8461 0.01591 *
## Residuals 27 10.4921  0.3886
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

unstack()

```
PlantGrowth.Tab <- unstack(PlantGrowth, weight ~ group)
PlantGrowth.Tab
```

```
##    ctrl trt1 trt2
## 1  4.17 4.81 6.31
## 2  5.58 4.17 5.12
## 3  5.18 4.41 5.54
## 4  6.11 3.59 5.50
## 5  4.50 5.87 5.37
## 6  4.61 3.83 5.29
## 7  5.17 6.03 4.92
## 8  4.53 4.89 6.15
## 9  5.33 4.32 5.80
## 10 5.14 4.69 5.26
```

```
write.csv(PlantGrowth.Tab, file = "PlantGrowthUnstacked.csv")
```

stack()

```
PlantGrowth.1 <- stack(PlantGrowth.Tab)
PlantGrowth.1
```

```
##    values  ind
## 1    4.17 ctrl
## 2    5.58 ctrl
## 3    5.18 ctrl
## 4    6.11 ctrl
## 5    4.50 ctrl
## 6    4.61 ctrl
## 7    5.17 ctrl
## 8    4.53 ctrl
## 9    5.33 ctrl
## 10   5.14 ctrl
## 11   4.81 trt1
## 12   4.17 trt1
## 13   4.41 trt1
## 14   3.59 trt1
## 15   5.87 trt1
## 16   3.83 trt1
## 17   6.03 trt1
## 18   4.89 trt1
## 19   4.32 trt1
## 20   4.69 trt1
## 21   6.31 trt2
## 22   5.12 trt2
## 23   5.54 trt2
## 24   5.50 trt2
## 25   5.37 trt2
## 26   5.29 trt2
## 27   4.92 trt2
## 28   6.15 trt2
## 29   5.80 trt2
## 30   5.26 trt2
```

```
names(PlantGrowth.1) <- c("weight", "group")

PlantGrowth.1 # return to original data
```

```
##    weight group
## 1    4.17  ctrl
## 2    5.58  ctrl
## 3    5.18  ctrl
## 4    6.11  ctrl
## 5    4.50  ctrl
## 6    4.61  ctrl
## 7    5.17  ctrl
## 8    4.53  ctrl
## 9    5.33  ctrl
## 10   5.14  ctrl
## 11   4.81  trt1
## 12   4.17  trt1
## 13   4.41  trt1
## 14   3.59  trt1
## 15   5.87  trt1
## 16   3.83  trt1
## 17   6.03  trt1
## 18   4.89  trt1
## 19   4.32  trt1
## 20   4.69  trt1
## 21   6.31  trt2
## 22   5.12  trt2
## 23   5.54  trt2
## 24   5.50  trt2
## 25   5.37  trt2
## 26   5.29  trt2
## 27   4.92  trt2
## 28   6.15  trt2
## 29   5.80  trt2
## 30   5.26  trt2
```

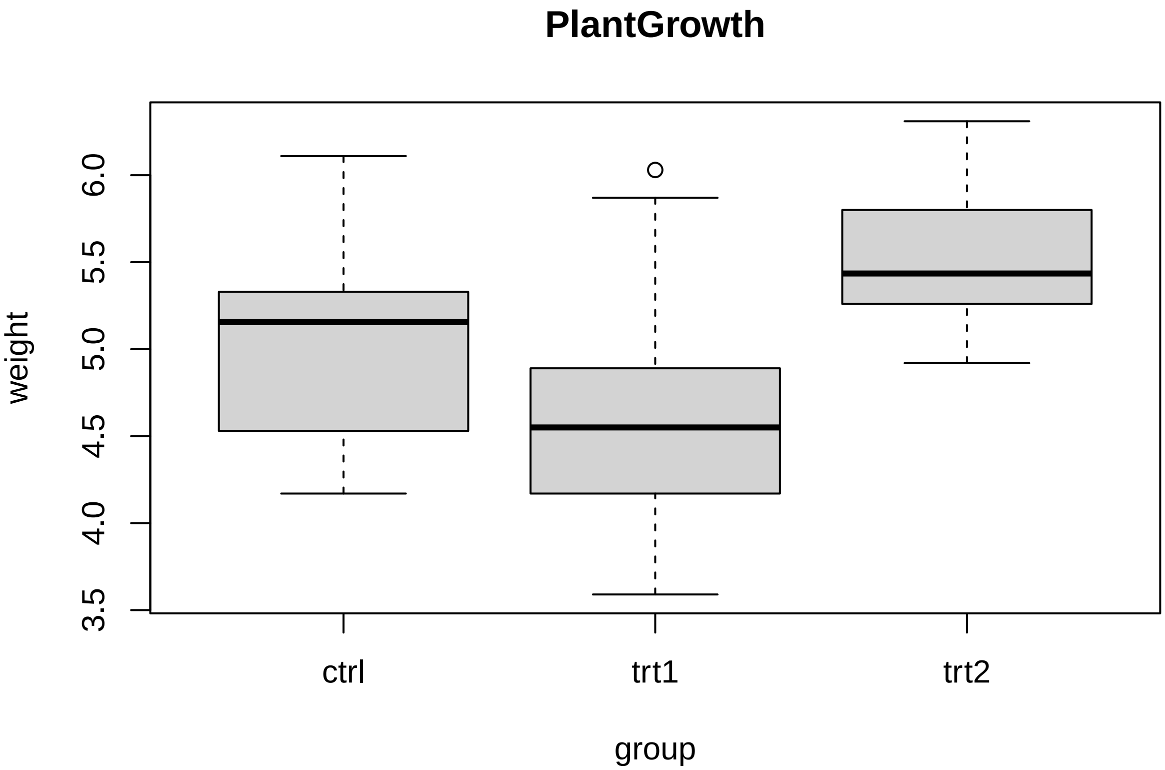
```
identical(PlantGrowth, PlantGrowth.1)
```

```
## [1] TRUE
```

Practice problem 1

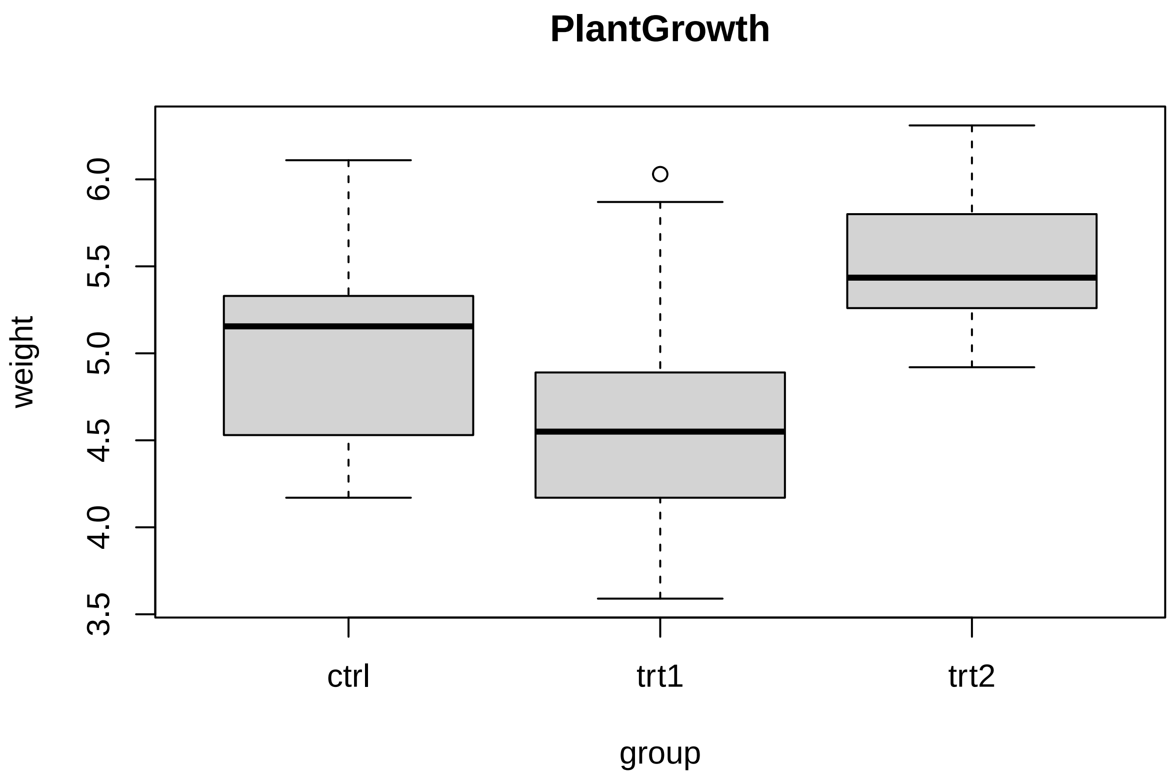
ex - box plot of PlantGrowth (weight ~ group)

```
with(PlantGrowth, boxplot(weight ~ group, main = "PlantGrowth"))
```



Then, whata about PlantGrowth.1?

```
boxplot(PlantGrowth.Tab,
        main = "PlantGrowth",
        xlab = "group",
        ylab= "weight")
```



2. Transpose

```
PlantGrowth.Tab
```

```
##    ctrl trt1 trt2
## 1  4.17 4.81 6.31
## 2  5.58 4.17 5.12
## 3  5.18 4.41 5.54
## 4  6.11 3.59 5.50
## 5  4.50 5.87 5.37
## 6  4.61 3.83 5.29
## 7  5.17 6.03 4.92
## 8  4.53 4.89 6.15
## 9  5.33 4.32 5.80
## 10 5.14 4.69 5.26
```

```
PlantGrowth.Tab.1 <- t(PlantGrowth.Tab)

PlantGrowth.Tab.1
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## ctrl 4.17 5.58 5.18 6.11 4.50 4.61 5.17 4.53 5.33  5.14
## trt1 4.81 4.17 4.41 3.59 5.87 3.83 6.03 4.89 4.32  4.69
## trt2 6.31 5.12 5.54 5.50 5.37 5.29 4.92 6.15 5.80  5.26
```

```
class(PlantGrowth.Tab.1)
```

```
## [1] "matrix" "array"
```

```
PlantGrowth.Tab.2 <- as.data.frame(PlantGrowth.Tab.1)
colnames(PlantGrowth.Tab.2) <- paste("rep", 1:10, sep = ".")
```

```
PlantGrowth.Tab.2
```

```
##      rep.1 rep.2 rep.3 rep.4 rep.5 rep.6 rep.7 rep.8 rep.9 rep.10
## ctrl  4.17  5.58  5.18  6.11  4.50  4.61  5.17  4.53  5.33   5.14
## trt1  4.81  4.17  4.41  3.59  5.87  3.83  6.03  4.89  4.32   4.69
## trt2  6.31  5.12  5.54  5.50  5.37  5.29  4.92  6.15  5.80   5.26
```

```
PlantGrowth.Tab.2$rep.10
```

```
## [1] 5.14 4.69 5.26
```

```
PlantGrowth.Tab.2[ , 10]
```

```
## [1] 5.14 4.69 5.26
```

3. array permutation

```
A <- matrix(1:12, c(4,3))
A
```

```
##      [,1] [,2] [,3]
## [1,]    1    5    9
## [2,]    2    6   10
## [3,]    3    7   11
## [4,]    4    8   12
```

```
is.matrix(A)
```

```
## [1] TRUE
```

```
is.array(A)
```

```
## [1] TRUE
```

We need to convert the dimension of specific array to lower dimension (ex - 3 to 2)

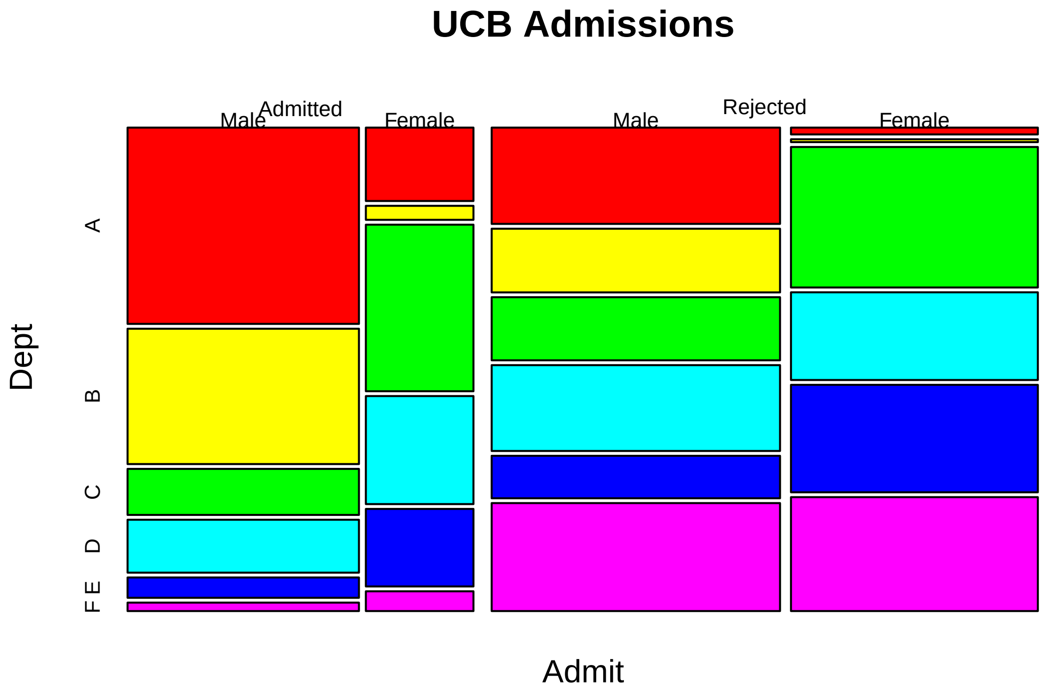
```
data(UCBAdmissions)

str(UCBAdmissions)
```

```
##  'table' num [1:2, 1:2, 1:6] 512 313 89 19 353 207 17 8 120 205 ...
##  - attr(*, "dimnames")=List of 3
##   ..$ Admit : chr [1:2] "Admitted" "Rejected"
##   ..$ Gender: chr [1:2] "Male" "Female"
##   ..$ Dept  : chr [1:6] "A" "B" "C" "D" ...
```

```
UCBAdmissions # 3 dimension array
```

```
## , , Dept = A
##
##           Gender
## Admit      Male Female
##   Admitted  512     89
##   Rejected  313     19
##
## , , Dept = B
##
##           Gender
## Admit      Male Female
##   Admitted  353     17
##   Rejected  207      8
##
## , , Dept = C
##
##           Gender
## Admit      Male Female
##   Admitted  120    202
##   Rejected  205    391
##
## , , Dept = D
##
##           Gender
## Admit      Male Female
##   Admitted  138    131
##   Rejected  279    244
##
## , , Dept = E
##
##           Gender
## Admit      Male Female
##   Admitted   53     94
##   Rejected  138    299
##
## , , Dept = F
##
##           Gender
## Admit      Male Female
##   Admitted   22     24
##   Rejected  351    317
```

```
mosaicplot(UCBAdmissions, off = c(2, 2, 5), dir = c("v", "v", "h"),
           col = rainbow(6), main = "UCB Admissions")
```



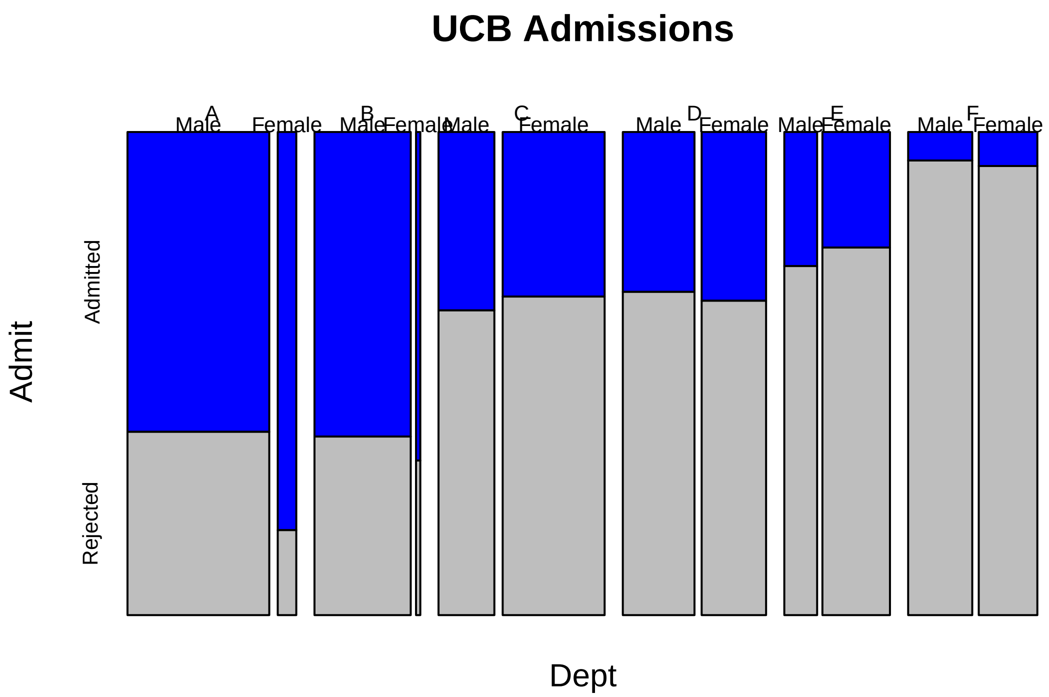
1st dimension: Admit -> Dept 2nd: Gender -> Gender 3rd: Dept -> Admit

aperm()

```
UCB.321 <- aperm(UCBAdmissions, c(3, 2, 1))
UCB.321
```

```
## , , Admit = Admitted
##
##     Gender
## Dept Male Female
##    A  512     89
##    B  353     17
##    C  120    202
##    D  138    131
##    E   53     94
##    F   22     24
##
## , , Admit = Rejected
##
##     Gender
## Dept Male Female
##    A  313     19
##    B  207      8
##    C  205    391
##    D  279    244
##    E  138    299
##    F  351    317
```

```
mosaicplot(UCB.321, off = c(10, 5, 0), dir = c("v", "v", "h"),
           col = c("blue", "gray"), main = "UCB Admissions")
```



```
# one of Simpson's paradox examples
```

Practice problem 2: transpose a matrix with aperm() function

```
A
```

```
##      [,1] [,2] [,3]
## [1,]    1    5    9
## [2,]    2    6   10
## [3,]    3    7   11
## [4,]    4    8   12
```

```
A.t1 <- t(A)
A.t2 <- aperm(A)

identical(A.t1, A.t2)
```

```
## [1] TRUE
```

4. reshaping: longitudinal data

```
data(Indometh)
set.seed(123)
Indometh$BP <- round(rnorm(nrow(Indometh), 100, 15))
Indometh
```

```
##    Subject time conc  BP
## 1        1 0.25 1.50  92
## 2        1 0.50 0.94  97
## 3        1 0.75 0.78 123
## 4        1 1.00 0.48 101
## 5        1 1.25 0.37 102
## 6        1 2.00 0.19 126
## 7        1 3.00 0.12 107
## 8        1 4.00 0.11  81
## 9        1 5.00 0.08  90
## 10       1 6.00 0.07  93
## 11       1 8.00 0.05 118
## 12       2 0.25 2.03 105
## 13       2 0.50 1.63 106
## 14       2 0.75 0.71 102
## 15       2 1.00 0.70  92
## 16       2 1.25 0.64 127
## 17       2 2.00 0.36 107
## 18       2 3.00 0.32  71
## 19       2 4.00 0.20 111
## 20       2 5.00 0.25  93
## 21       2 6.00 0.12  84
## 22       2 8.00 0.08  97
## 23       3 0.25 2.72  85
## 24       3 0.50 1.49  89
## 25       3 0.75 1.16  91
## 26       3 1.00 0.80  75
## 27       3 1.25 0.80 113
## 28       3 2.00 0.39 102
## 29       3 3.00 0.22  83
## 30       3 4.00 0.12 119
## 31       3 5.00 0.11 106
## 32       3 6.00 0.08  96
## 33       3 8.00 0.08 113
## 34       4 0.25 1.85 113
## 35       4 0.50 1.39 112
## 36       4 0.75 1.02 110
## 37       4 1.00 0.89 108
## 38       4 1.25 0.59  99
## 39       4 2.00 0.40  95
## 40       4 3.00 0.16  94
## 41       4 4.00 0.11  90
## 42       4 5.00 0.10  97
## 43       4 6.00 0.07  81
## 44       4 8.00 0.07 133
## 45       5 0.25 2.05 118
## 46       5 0.50 1.04  83
## 47       5 0.75 0.81  94
## 48       5 1.00 0.39  93
## 49       5 1.25 0.30 112
## 50       5 2.00 0.23  99
## 51       5 3.00 0.13 104
## 52       5 4.00 0.11 100
## 53       5 5.00 0.08  99
## 54       5 6.00 0.10 121
## 55       5 8.00 0.06  97
## 56       6 0.25 2.31 123
## 57       6 0.50 1.44  77
## 58       6 0.75 1.03 109
## 59       6 1.00 0.84 102
## 60       6 1.25 0.64 103
## 61       6 2.00 0.42 106
## 62       6 3.00 0.24  92
## 63       6 4.00 0.17  95
## 64       6 5.00 0.13  85
## 65       6 6.00 0.10  84
## 66       6 8.00 0.09 105
```

```
Indometh.wide <- reshape(Indometh, idvar = "Subject",
                         v.names = c("conc", "BP"), timevar = "time",
                         sep = "_", direction = "wide")
Indometh.wide
```

```
##    Subject conc_0.25 BP_0.25 conc_0.5 BP_0.5 conc_0.75 BP_0.75 conc_1 BP_1
## 1        1      1.50      92     0.94     97      0.78     123   0.48  101
## 12       2      2.03     105     1.63    106      0.71     102   0.70   92
## 23       3      2.72      85     1.49     89      1.16      91   0.80   75
## 34       4      1.85     113     1.39    112      1.02     110   0.89  108
## 45       5      2.05     118     1.04     83      0.81      94   0.39   93
## 56       6      2.31     123     1.44     77      1.03     109   0.84  102
##    conc_1.25 BP_1.25 conc_2 BP_2 conc_3 BP_3 conc_4 BP_4 conc_5 BP_5 conc_6
## 1       0.37     102   0.19  126   0.12  107   0.11   81   0.08   90   0.07
## 12      0.64     127   0.36  107   0.32   71   0.20  111   0.25   93   0.12
## 23      0.80     113   0.39  102   0.22   83   0.12  119   0.11  106   0.08
## 34      0.59      99   0.40   95   0.16   94   0.11   90   0.10   97   0.07
## 45      0.30     112   0.23   99   0.13  104   0.11  100   0.08   99   0.10
## 56      0.64     103   0.42  106   0.24   92   0.17   95   0.13   85   0.10
##    BP_6 conc_8 BP_8
## 1    93   0.05  118
## 12   84   0.08   97
## 23   96   0.08  113
## 34   81   0.07  133
## 45  121   0.06   97
## 56   84   0.09  105
```

```
Indometh.wide.1 <- data.frame(Indometh.wide)
Indometh.wide.1 <- reshape(Indometh.wide.1,
                           idvar = "Subject",
                           varying = colnames(Indometh.wide.1)[-1],
                           sep = "_",
                           direction = "long")
```

practice problem 3. take a look at the class of Indometh.wide and compare it with original one

```
class(Indometh.wide)
```

```
## [1] "nfnGroupedData" "nfGroupedData"  "groupedData"    "data.frame"
```

```
class(Indometh)
```

```
## [1] "nfnGroupedData" "nfGroupedData"  "groupedData"    "data.frame"
```

practice problem 3. take a look at the class of Indometh.wide and compare it with original one

```
Indometh.conc <- unstack(Indometh, conc ~ Subject)
Indometh.conc <- Indometh.conc[, order(colnames(Indometh.conc))]
Indometh.conc <- t(Indometh.conc)
colnames(Indometh.conc) <- paste("conc", levels(as.factor(Indometh$time)), sep = "_")

Indometh.BP <- unstack(Indometh, BP ~ Subject)
Indometh.BP <- Indometh.BP[, order(colnames(Indometh.BP))]
Indometh.BP <- t(Indometh.BP)
colnames(Indometh.BP) <- paste("BP", levels(as.factor(Indometh$time)), sep = "_")

Indometh.new <- data.frame(Indometh.conc, Indometh.BP)
Indometh.new <- Indometh.new[, order(sapply(strsplit(colnames(Indometh.new), "_", fixed = T), "[", 2))]
Indometh.new$Subject <- as.factor(1:6)
Indometh.new <- Indometh.new[, c(23, 1:22)]

Indometh.new
```

```
##    Subject conc_0.25 BP_0.25 conc_0.5 BP_0.5 conc_0.75 BP_0.75 conc_1 BP_1
## X1       1      1.50      92     0.94     97      0.78     123   0.48  101
## X2       2      2.03     105     1.63    106      0.71     102   0.70   92
## X3       3      2.72      85     1.49     89      1.16      91   0.80   75
## X4       4      1.85     113     1.39    112      1.02     110   0.89  108
## X5       5      2.05     118     1.04     83      0.81      94   0.39   93
## X6       6      2.31     123     1.44     77      1.03     109   0.84  102
##    conc_1.25 BP_1.25 conc_2 BP_2 conc_3 BP_3 conc_4 BP_4 conc_5 BP_5 conc_6
## X1      0.37     102   0.19  126   0.12  107   0.11   81   0.08   90   0.07
## X2      0.64     127   0.36  107   0.32   71   0.20  111   0.25   93   0.12
## X3      0.80     113   0.39  102   0.22   83   0.12  119   0.11  106   0.08
## X4      0.59      99   0.40   95   0.16   94   0.11   90   0.10   97   0.07
## X5      0.30     112   0.23   99   0.13  104   0.11  100   0.08   99   0.10
## X6      0.64     103   0.42  106   0.24   92   0.17   95   0.13   85   0.10
##    BP_6 conc_8 BP_8
## X1   93   0.05  118
## X2   84   0.08   97
## X3   96   0.08  113
## X4   81   0.07  133
## X5  121   0.06   97
## X6   84   0.09  105
```