Functions.R

sten

Tue Feb 26 08:52:42 2019

```
# First we make some example data

library(JM)
```

```
## Loading required package: MASS
```

```
## Loading required package: nlme
```

```
## Loading required package: splines
```

```
## Loading required package: survival
```

```
set.seed(2015+1)
patient <- c(1:20)
name <- paste('Patient', LETTERS[1:20])
height <- rnorm(20, 1.70, 0.1)
weight <- rnorm(20, 70, 10)
sex <- sample(0:1, 20, replace = TRUE)
sex <- factor(sex, levels = 0:1, labels = c("male", "female"))

dat <- data.frame(patient, name, height, weight, sex)
```

Functions

```
## How many patients do we have in the "dat" dataset
length(dat$patient)
```

```
## [1] 20
```

```
#take care
length(dat)
```

```
## [1] 5
```

```
dim(dat)
```

```
## [1] 20  5
```

```
nrow(dat)
```

```
## [1] 20
```

```
#however
nrow(dat[,c(1)])
```

```
## NULL
```

```
NROW(dat[,c(1)])
```

```
## [1] 20
```

```
M<-matrix(1:9,3)
nrow(M)
```

```
## [1] 3
```

```
## Summation of height
sum(dat$height)
```

```
## [1] 33.67665
```

```
## Mean of weight and height
mean(dat$weight)
```

```
## [1] 67.77676
```

```
mean(dat$height)
```

```
## [1] 1.683833
```

```
apply(dat[,3:4], 2, mean) # explained later
```

```
##    height    weight
##  1.683833 67.776760
```

```
## Median of height
median(dat$weight)
```

```
## [1] 65.42157
```

```
## Quantiles of height
quantile(dat$weight)
```

```
##       0%      25%      50%      75%     100%
## 54.98204 61.22761 65.42157 72.83606 83.53277
```

```
quantile(dat$weight, probs = c(0.333, 0.667))
```

```
##    33.3%    66.7%
## 62.43739 70.08140
```

```
## Variance of weight
var(dat$weight)
```

```
## [1] 85.28777
```

```
## Standard deviation of weight
sd(dat$weight)
```

```
## [1] 9.235138
```

```
## Min of height
min(dat$height)
```

```
## [1] 1.420853
```

```
## Max of height
max(dat$height)
```

```
## [1] 1.859492
```

```
## Range of height
range(dat$height)
```

```
## [1] 1.420853 1.859492
```

```
## Calculate the mean weight
ave(dat$weight)
```

```
##  [1] 67.77676 67.77676 67.77676 67.77676 67.77676 67.77676 67.77676
##  [8] 67.77676 67.77676 67.77676 67.77676 67.77676 67.77676 67.77676
## [15] 67.77676 67.77676 67.77676 67.77676 67.77676 67.77676
```

```
ave(dat$weight, FUN=median)
```

```
##  [1] 65.42157 65.42157 65.42157 65.42157 65.42157 65.42157 65.42157
##  [8] 65.42157 65.42157 65.42157 65.42157 65.42157 65.42157 65.42157
## [15] 65.42157 65.42157 65.42157 65.42157 65.42157 65.42157
```

```
(dat$grpmed <-ave(dat$weight, dat$sex, FUN=median))
```

```
##  [1] 64.28269 64.28269 67.95147 67.95147 64.28269 64.28269 64.28269
##  [8] 67.95147 67.95147 67.95147 64.28269 67.95147 64.28269 67.95147
## [15] 67.95147 64.28269 67.95147 67.95147 64.28269 64.28269
```

```
## Cumulative summation of weight
cumsum(dat$weight)
```

```
##  [1]   65.80858  129.26105  196.01217  271.96573  343.76263  403.72783
##  [7]  467.25865  550.73387  632.04567  691.28420  751.35047  820.50227
## [13]  902.80205  957.78409 1028.31716 1111.84993 1173.46466 1235.40883
## [19] 1300.44339 1355.53520
```

```
## Log of height
log(dat$height)
```

```
##  [1] 0.4753181 0.5878560 0.5273037 0.5479275 0.3512573 0.5138566 0.4846762
##  [8] 0.4894992 0.5519910 0.5413352 0.5048971 0.5139240 0.6064508 0.4758869
## [15] 0.6203033 0.5500753 0.4776560 0.5900707 0.4884655 0.4886696
```

```
log(dat$height, base = exp(1))
```

```
##  [1] 0.4753181 0.5878560 0.5273037 0.5479275 0.3512573 0.5138566 0.4846762
##  [8] 0.4894992 0.5519910 0.5413352 0.5048971 0.5139240 0.6064508 0.4758869
## [15] 0.6203033 0.5500753 0.4776560 0.5900707 0.4884655 0.4886696
```

```
log10(dat$height) # base 10
```

```
##  [1] 0.2064280 0.2553026 0.2290051 0.2379619 0.1525491 0.2231651 0.2104922
##  [8] 0.2125868 0.2397267 0.2350989 0.2192740 0.2231943 0.2633782 0.2066751
## [15] 0.2693943 0.2388947 0.2074434 0.2562645 0.2121379 0.2122265
```

```
log2(dat$height) # base 2
```

```
##  [1] 0.6857391 0.8480969 0.7607385 0.7904922 0.5067572 0.7413384 0.6992399
##  [8] 0.7061981 0.7963547 0.7809817 0.7284125 0.7414356 0.8749236 0.6865597
## [15] 0.8949085 0.7935909 0.6891119 0.8512921 0.7047067 0.7050012
```

```
## Exp of height
exp(dat$height)
```

```
##  [1] 4.995442 6.050402 5.443149 5.638762 4.140651 5.321344 5.071563
##  [8] 5.111531 5.678615 5.575041 5.242586 5.321944 6.258315 5.000015
## [15] 6.420474 5.659772 5.014285 6.074599 5.102922 5.104620
```

```
## Calculate the mean of weight per sex and weight group
dat$weight_65higher <- as.numeric(dat$weight > 65)
dat$weight_65higher <- factor(dat$weight_65higher, levels = c(0:1),
                              labels = c("low","high"))
dat$weight_65higher
```

```
##  [1] high low  high high high low  low  high high low  low  high high low
## [15] high high low  low  high low
## Levels: low high
```

```
dat$weight_65higher <- (as.numeric(dat$weight > 65) + 3)
dat$weight_65higher <- factor(dat$weight_65higher, levels = c(3:4),
                              labels = c("low","high"))
dat$weight_65higher
```

```
##  [1] high low  high high high low  low  high high low  low  high high low
## [15] high high low  low  high low
## Levels: low high
```

```
tapply(dat$weight, list(dat$sex, dat$weight_65higher), mean)
```

```
##             low     high
## male   60.42131 73.69451
## female 59.44487 74.52943
```

```
## Sort the values of height
sort(dat$height)
```

```
##  [1] 1.420853 1.608526 1.609441 1.612291 1.623649 1.629813 1.630146
##  [8] 1.631499 1.656815 1.671726 1.671839 1.694358 1.718300 1.729665
## [15] 1.733384 1.736707 1.800125 1.804116 1.833911 1.859492
```

```
## Reverse the values of height
rev(dat$height)
```

```
##  [1] 1.630146 1.629813 1.804116 1.612291 1.733384 1.859492 1.609441
##  [8] 1.833911 1.671839 1.656815 1.718300 1.736707 1.631499 1.623649
## [15] 1.671726 1.420853 1.729665 1.694358 1.800125 1.608526
```

```
dat[order(dat$weight), ]
```

```
##    patient      name   height   weight    sex   grpmed weight_65higher
## 14      14 Patient N 1.609441 54.98204 female 67.95147             low
## 20      20 Patient T 1.630146 55.09181   male 64.28269             low
## 10      10 Patient J 1.718300 59.23853 female 67.95147             low
## 6        6 Patient F 1.671726 59.96520   male 64.28269             low
## 11      11 Patient K 1.656815 60.06626   male 64.28269             low
## 17      17 Patient Q 1.612291 61.61473 female 67.95147             low
## 18      18 Patient R 1.804116 61.94418 female 67.95147             low
## 2        2 Patient B 1.800125 63.45247   male 64.28269             low
## 7        7 Patient G 1.623649 63.53082   male 64.28269             low
## 19      19 Patient S 1.629813 65.03455   male 64.28269            high
## 1        1 Patient A 1.608526 65.80858   male 64.28269            high
## 3        3 Patient C 1.694358 66.75113 female 67.95147            high
## 12      12 Patient L 1.671839 69.15181 female 67.95147            high
## 15      15 Patient O 1.859492 70.53307 female 67.95147            high
## 5        5 Patient E 1.420853 71.79690   male 64.28269            high
## 4        4 Patient D 1.729665 75.95356 female 67.95147            high
```

```
## 9        9 Patient I 1.736707 81.31180 female 67.95147            high
## 13      13 Patient M 1.833911 82.29977   male 64.28269            high
## 8        8 Patient H 1.631499 83.47523 female 67.95147            high
## 16      16 Patient P 1.733384 83.53277   male 64.28269            high
```

```
dat[order(dat$weight, dat$height), ]
```

```
##    patient      name   height   weight    sex   grpmed weight_65higher
## 14      14 Patient N 1.609441 54.98204 female 67.95147             low
## 20      20 Patient T 1.630146 55.09181   male 64.28269             low
## 10      10 Patient J 1.718300 59.23853 female 67.95147             low
## 6        6 Patient F 1.671726 59.96520   male 64.28269             low
## 11      11 Patient K 1.656815 60.06626   male 64.28269             low
## 17      17 Patient Q 1.612291 61.61473 female 67.95147             low
## 18      18 Patient R 1.804116 61.94418 female 67.95147             low
## 2        2 Patient B 1.800125 63.45247   male 64.28269             low
## 7        7 Patient G 1.623649 63.53082   male 64.28269             low
## 19      19 Patient S 1.629813 65.03455   male 64.28269            high
## 1        1 Patient A 1.608526 65.80858   male 64.28269            high
## 3        3 Patient C 1.694358 66.75113 female 67.95147            high
## 12      12 Patient L 1.671839 69.15181 female 67.95147            high
## 15      15 Patient O 1.859492 70.53307 female 67.95147            high
## 5        5 Patient E 1.420853 71.79690   male 64.28269            high
## 4        4 Patient D 1.729665 75.95356 female 67.95147            high
## 9        9 Patient I 1.736707 81.31180 female 67.95147            high
## 13      13 Patient M 1.833911 82.29977   male 64.28269            high
## 8        8 Patient H 1.631499 83.47523 female 67.95147            high
## 16      16 Patient P 1.733384 83.53277   male 64.28269            high
```

```
dat[order(dat$weight, -dat$height), ]
```

```
##    patient      name   height   weight    sex   grpmed weight_65higher
## 14      14 Patient N 1.609441 54.98204 female 67.95147             low
## 20      20 Patient T 1.630146 55.09181   male 64.28269             low
## 10      10 Patient J 1.718300 59.23853 female 67.95147             low
## 6        6 Patient F 1.671726 59.96520   male 64.28269             low
## 11      11 Patient K 1.656815 60.06626   male 64.28269             low
## 17      17 Patient Q 1.612291 61.61473 female 67.95147             low
## 18      18 Patient R 1.804116 61.94418 female 67.95147             low
## 2        2 Patient B 1.800125 63.45247   male 64.28269             low
## 7        7 Patient G 1.623649 63.53082   male 64.28269             low
## 19      19 Patient S 1.629813 65.03455   male 64.28269            high
## 1        1 Patient A 1.608526 65.80858   male 64.28269            high
## 3        3 Patient C 1.694358 66.75113 female 67.95147            high
## 12      12 Patient L 1.671839 69.15181 female 67.95147            high
## 15      15 Patient O 1.859492 70.53307 female 67.95147            high
## 5        5 Patient E 1.420853 71.79690   male 64.28269            high
## 4        4 Patient D 1.729665 75.95356 female 67.95147            high
## 9        9 Patient I 1.736707 81.31180 female 67.95147            high
## 13      13 Patient M 1.833911 82.29977   male 64.28269            high
## 8        8 Patient H 1.631499 83.47523 female 67.95147            high
## 16      16 Patient P 1.733384 83.53277   male 64.28269            high
```

```
dat[order(dat$weight, dat$height,
          decreasing=c(FALSE, TRUE)), ]
```

```
##    patient      name   height   weight    sex   grpmed weight_65higher
## 14      14 Patient N 1.609441 54.98204 female 67.95147             low
## 20      20 Patient T 1.630146 55.09181   male 64.28269             low
## 10      10 Patient J 1.718300 59.23853 female 67.95147             low
## 6        6 Patient F 1.671726 59.96520   male 64.28269             low
## 11      11 Patient K 1.656815 60.06626   male 64.28269             low
## 17      17 Patient Q 1.612291 61.61473 female 67.95147             low
## 18      18 Patient R 1.804116 61.94418 female 67.95147             low
## 2        2 Patient B 1.800125 63.45247   male 64.28269             low
## 7        7 Patient G 1.623649 63.53082   male 64.28269             low
## 19      19 Patient S 1.629813 65.03455   male 64.28269            high
## 1        1 Patient A 1.608526 65.80858   male 64.28269            high
## 3        3 Patient C 1.694358 66.75113 female 67.95147            high
## 12      12 Patient L 1.671839 69.15181 female 67.95147            high
## 15      15 Patient O 1.859492 70.53307 female 67.95147            high
## 5        5 Patient E 1.420853 71.79690   male 64.28269            high
## 4        4 Patient D 1.729665 75.95356 female 67.95147            high
## 9        9 Patient I 1.736707 81.31180 female 67.95147            high
## 13      13 Patient M 1.833911 82.29977   male 64.28269            high
## 8        8 Patient H 1.631499 83.47523 female 67.95147            high
## 16      16 Patient P 1.733384 83.53277   male 64.28269            high
```

```
rank(dat$weight)
```

```
##  [1] 11  8 12 16 15  4  9 19 17  3  5 13 18  1 14 20  6  7 10  2
```

```
## Check for duplicates
duplicated(dat$weight)
```

```
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [12] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
duplicated(c( 1, 1, 1, 2, 2, 1))
```

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

```
unique(c(1,4,1,2,3, 1)) # unique values
```

```
## [1] 1 4 2 3
```

```
## Split the dataset for males and females
split(dat, dat$sex)
```

```
## $male
##    patient      name   height   weight  sex   grpmed weight_65higher
## 1        1 Patient A 1.608526 65.80858 male 64.28269            high
## 2        2 Patient B 1.800125 63.45247 male 64.28269             low
## 5        5 Patient E 1.420853 71.79690 male 64.28269            high
## 6        6 Patient F 1.671726 59.96520 male 64.28269             low
## 7        7 Patient G 1.623649 63.53082 male 64.28269             low
## 11      11 Patient K 1.656815 60.06626 male 64.28269             low
## 13      13 Patient M 1.833911 82.29977 male 64.28269            high
## 16      16 Patient P 1.733384 83.53277 male 64.28269            high
## 19      19 Patient S 1.629813 65.03455 male 64.28269            high
## 20      20 Patient T 1.630146 55.09181 male 64.28269             low
##
## $female
##    patient      name   height   weight    sex   grpmed weight_65higher
```

```
## 3        3 Patient C 1.694358 66.75113 female 67.95147           high
## 4        4 Patient D 1.729665 75.95356 female 67.95147           high
## 8        8 Patient H 1.631499 83.47523 female 67.95147           high
## 9        9 Patient I 1.736707 81.31180 female 67.95147           high
## 10      10 Patient J 1.718300 59.23853 female 67.95147            low
## 12      12 Patient L 1.671839 69.15181 female 67.95147           high
## 14      14 Patient N 1.609441 54.98204 female 67.95147            low
## 15      15 Patient O 1.859492 70.53307 female 67.95147           high
## 17      17 Patient Q 1.612291 61.61473 female 67.95147            low
## 18      18 Patient R 1.804116 61.94418 female 67.95147            low
```

```
## Split weight in two intervals
cut(dat$weight, 2)
```

```
##  [1] (55,69.3]   (55,69.3]   (55,69.3]   (69.3,83.6] (69.3,83.6]
##  [6] (55,69.3]   (55,69.3]   (69.3,83.6] (69.3,83.6] (55,69.3]
## [11] (55,69.3]   (55,69.3]   (69.3,83.6] (55,69.3]   (69.3,83.6]
## [16] (69.3,83.6] (55,69.3]   (55,69.3]   (55,69.3]   (55,69.3]
## Levels: (55,69.3] (69.3,83.6]
```

```
cut(dat$weight, c(54, 70, 85))
```

```
##  [1] (54,70] (54,70] (54,70] (70,85] (70,85] (54,70] (54,70] (70,85]
##  [9] (70,85] (54,70] (54,70] (54,70] (70,85] (54,70] (70,85] (70,85]
## [17] (54,70] (54,70] (54,70] (54,70]
## Levels: (54,70] (70,85]
```

```
cut(dat$weight, 3)
```

```
##  [1] (64.5,74] (55,64.5] (64.5,74] (74,83.6] (64.5,74] (55,64.5] (55,64.5]
##  [8] (74,83.6] (74,83.6] (55,64.5] (55,64.5] (64.5,74] (74,83.6] (55,64.5]
## [15] (64.5,74] (74,83.6] (55,64.5] (55,64.5] (64.5,74] (55,64.5]
## Levels: (55,64.5] (64.5,74] (74,83.6]
```

```
## Obtain frequencies
table(dat$weight_65higher)
```

```
##
##  low high
##    9   11
```

```
## Tables
table(dat$weight_65higher, dat$sex)
```

```
##
##        male female
##   low     5      4
##   high    5      6
```

```
dat$BMI <- dat$weight/(dat$height^2)
dat$BMIcat <- as.numeric(cut(dat$BMI, 2))
ftable(xtabs(~ dat$weight_65higher + dat$sex + dat$BMIcat))
```

```
##                             dat$BMIcat 1 2
## dat$weight_65higher dat$sex
## low                 male               5 0
##                     female             4 0
## high                male               3 2
##                     female             5 1
```

```r
## Exclude patients with missing values in more than one covariates
dat1 <- dat
dat1$weight[2] <- NA
dat1 <- dat1[complete.cases(dat1$weight, dat1$height, dat1$sex), ]
dat <- dat[complete.cases(dat$weight, dat$height, dat$sex), ]

### General functions
## repeat values
rep(1:2, 2)
```

```
## [1] 1 2 1 2
```

```r
rep(1:2, time = 2)
```

```
## [1] 1 2 1 2
```

```r
rep(1:2, each = 2)
```

```
## [1] 1 1 2 2
```

```r
rep(mean(dat$weight), 20)
```

```
##  [1] 67.77676 67.77676 67.77676 67.77676 67.77676 67.77676 67.77676
##  [8] 67.77676 67.77676 67.77676 67.77676 67.77676 67.77676 67.77676
## [15] 67.77676 67.77676 67.77676 67.77676 67.77676 67.77676
```

```r
## Generate sequences
2:20
```

```
##  [1]  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
```

```r
seq(1, 10, by = 2)
```

```
## [1] 1 3 5 7 9
```

```r
seq(1,10, length.out=100)
```

```
##   [1]  1.000000  1.090909  1.181818  1.272727  1.363636  1.454545  1.545455
##   [8]  1.636364  1.727273  1.818182  1.909091  2.000000  2.090909  2.181818
##  [15]  2.272727  2.363636  2.454545  2.545455  2.636364  2.727273  2.818182
##  [22]  2.909091  3.000000  3.090909  3.181818  3.272727  3.363636  3.454545
##  [29]  3.545455  3.636364  3.727273  3.818182  3.909091  4.000000  4.090909
##  [36]  4.181818  4.272727  4.363636  4.454545  4.545455  4.636364  4.727273
##  [43]  4.818182  4.909091  5.000000  5.090909  5.181818  5.272727  5.363636
##  [50]  5.454545  5.545455  5.636364  5.727273  5.818182  5.909091  6.000000
##  [57]  6.090909  6.181818  6.272727  6.363636  6.454545  6.545455  6.636364
##  [64]  6.727273  6.818182  6.909091  7.000000  7.090909  7.181818  7.272727
##  [71]  7.363636  7.454545  7.545455  7.636364  7.727273  7.818182  7.909091
##  [78]  8.000000  8.090909  8.181818  8.272727  8.363636  8.454545  8.545455
##  [85]  8.636364  8.727273  8.818182  8.909091  9.000000  9.090909  9.181818
##  [92]  9.272727  9.363636  9.454545  9.545455  9.636364  9.727273  9.818182
##  [99]  9.909091 10.000000
```

```r
ind <- seq(min(dat$patient), max(dat$patient), by = 2)

dat[ind, ]
```

```
##    patient      name   height   weight    sex   grpmed weight_65higher
## 1        1 Patient A 1.608526 65.80858   male 64.28269            high
```

```
## 3        3 Patient C 1.694358 66.75113 female 67.95147           high
## 5        5 Patient E 1.420853 71.79690   male 64.28269           high
## 7        7 Patient G 1.623649 63.53082   male 64.28269            low
## 9        9 Patient I 1.736707 81.31180 female 67.95147           high
## 11      11 Patient K 1.656815 60.06626   male 64.28269            low
## 13      13 Patient M 1.833911 82.29977   male 64.28269           high
## 15      15 Patient O 1.859492 70.53307 female 67.95147           high
## 17      17 Patient Q 1.612291 61.61473 female 67.95147            low
## 19      19 Patient S 1.629813 65.03455   male 64.28269           high
##         BMI BMIcat
## 1  25.43469      1
## 3  23.25136      1
## 5  35.56374      2
## 7  24.09906      1
## 9  26.95878      1
## 11 21.88178      1
## 13 24.47046      1
## 15 20.39878      1
## 17 23.70270      1
## 19 24.48322      1
```

```
## For character vectors
ch1 <- letters[1:6]
ch2 <- LETTERS[1:6]
ch1
```

```
## [1] "a" "b" "c" "d" "e" "f"
```

```
ch2
```

```
## [1] "A" "B" "C" "D" "E" "F"
```

```
toupper(ch1) # to upper case
```

```
## [1] "A" "B" "C" "D" "E" "F"
```

```
tolower(ch2) # to lower case
```

```
## [1] "a" "b" "c" "d" "e" "f"
```

```
paste(ch1, ch2)
```

```
## [1] "a A" "b B" "c C" "d D" "e E" "f F"
```

```
paste(ch1, ch2, sep = "-")
```

```
## [1] "a-A" "b-B" "c-C" "d-D" "e-E" "f-F"
```

```
paste(ch1, ch2, collapse = ",")
```

```
## [1] "a A,b B,c C,d D,e E,f F"
```

```
paste("Hello", "world")
```

```
## [1] "Hello world"
```

```
paste("Hello", "world", sep = "")
```

```
## [1] "Helloworld"
```

```
paste0("Hello", "world")
```

```
## [1] "Helloworld"
```

```
## Dimensions
dim(dat)
```

```
## [1] 20  9
```

```
ncol(dat)
```

```
## [1] 9
```

```
nrow(dat)
```

```
## [1] 20
```

```
## Obtain diagonal
mat <- matrix(1:9, 3, 3)
diag(mat)
```

```
## [1] 1 5 9
```

```
diag(5) # create an identity matrix
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    0    0    0    0
## [2,]    0    1    0    0    0
## [3,]    0    0    1    0    0
## [4,]    0    0    0    1    0
## [5,]    0    0    0    0    1
```

```
diag(1:3) # create a diagonal matrix
```

```
##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    0    2    0
## [3,]    0    0    3
```

```
## Compute the row sums
rowSums(mat) # sum values in each row / sum of the columns
```

```
## [1] 12 15 18
```

```
colSums(mat)
```

```
## [1]  6 15 24
```

```
## Matrix Transpose
t(mat)
```

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

```
## arithmetics
mat + mat
```

```
##      [,1] [,2] [,3]
## [1,]    2    8   14
## [2,]    4   10   16
## [3,]    6   12   18
```

```r
mat - mat
```

```
##      [,1] [,2] [,3]
## [1,]    0    0    0
## [2,]    0    0    0
## [3,]    0    0    0
```

```r
mat %*% mat
```

```
##      [,1] [,2] [,3]
## [1,]   30   66  102
## [2,]   36   81  126
## [3,]   42   96  150
```

```r
det(mat) # matrix determinant
```

```
## [1] 0
```

```r
solve(matrix(c(2,1,3,1),2)) # matrix inverse
```

```
##      [,1] [,2]
## [1,]   -1    3
## [2,]    1   -2
```

```r
var(matrix(c(1,2,1,3,4,6,7,12,8,9),5)) # variance-covariance matrix
```

```
##      [,1] [,2]
## [1,]  1.7 -0.1
## [2,] -0.1  5.3
```

```r
cor(matrix(c(1,2,1,3,4,6,7,12,8,9),5)) # correlation matrix
```

```
##             [,1]        [,2]
## [1,]  1.00000000 -0.03331483
## [2,] -0.03331483  1.00000000
```

```r
## Obtain all possible combinations
groups <- list(gp1 = 1:3, gp2 = 4:5, gp3 = 6:7, gp4 = 8:10, gp5 = 11)
expand.grid(groups$gp1, groups$gp2)
```

```
##   Var1 Var2
## 1    1    4
## 2    2    4
## 3    3    4
## 4    1    5
## 5    2    5
## 6    3    5
```

```r
expand.grid(groups)
```

```
##    gp1 gp2 gp3 gp4 gp5
## 1    1   4   6   8  11
## 2    2   4   6   8  11
## 3    3   4   6   8  11
## 4    1   5   6   8  11
## 5    2   5   6   8  11
## 6    3   5   6   8  11
## 7    1   4   7   8  11
## 8    2   4   7   8  11
## 9    3   4   7   8  11
```

```
## 10   1   5   7   8  11
## 11   2   5   7   8  11
## 12   3   5   7   8  11
## 13   1   4   6   9  11
## 14   2   4   6   9  11
## 15   3   4   6   9  11
## 16   1   5   6   9  11
## 17   2   5   6   9  11
## 18   3   5   6   9  11
## 19   1   4   7   9  11
## 20   2   4   7   9  11
## 21   3   4   7   9  11
## 22   1   5   7   9  11
## 23   2   5   7   9  11
## 24   3   5   7   9  11
## 25   1   4   6  10  11
## 26   2   4   6  10  11
## 27   3   4   6  10  11
## 28   1   5   6  10  11
## 29   2   5   6  10  11
## 30   3   5   6  10  11
## 31   1   4   7  10  11
## 32   2   4   7  10  11
## 33   3   4   7  10  11
## 34   1   5   7  10  11
## 35   2   5   7  10  11
## 36   3   5   7  10  11
```

```
cond1 <- c("Type 1", "Type 2", "Type 3")
cond2 <- c("Mild", "Moderate", "Severe")
cond3 <- c("Placebo", "Active")
expand.grid(cond1, cond2, cond3)
```

```
##      Var1     Var2    Var3
## 1  Type 1     Mild Placebo
## 2  Type 2     Mild Placebo
## 3  Type 3     Mild Placebo
## 4  Type 1 Moderate Placebo
## 5  Type 2 Moderate Placebo
## 6  Type 3 Moderate Placebo
## 7  Type 1   Severe Placebo
## 8  Type 2   Severe Placebo
## 9  Type 3   Severe Placebo
## 10 Type 1     Mild  Active
## 11 Type 2     Mild  Active
## 12 Type 3     Mild  Active
## 13 Type 1 Moderate  Active
## 14 Type 2 Moderate  Active
## 15 Type 3 Moderate  Active
## 16 Type 1   Severe  Active
## 17 Type 2   Severe  Active
## 18 Type 3   Severe  Active
```

```
## Combine vectors by column
cbind(dat$weight, dat$height)
```

```
##           [,1]     [,2]
##  [1,] 65.80858 1.608526
##  [2,] 63.45247 1.800125
##  [3,] 66.75113 1.694358
##  [4,] 75.95356 1.729665
##  [5,] 71.79690 1.420853
##  [6,] 59.96520 1.671726
##  [7,] 63.53082 1.623649
##  [8,] 83.47523 1.631499
##  [9,] 81.31180 1.736707
## [10,] 59.23853 1.718300
## [11,] 60.06626 1.656815
## [12,] 69.15181 1.671839
## [13,] 82.29977 1.833911
## [14,] 54.98204 1.609441
## [15,] 70.53307 1.859492
## [16,] 83.53277 1.733384
## [17,] 61.61473 1.612291
## [18,] 61.94418 1.804116
## [19,] 65.03455 1.629813
## [20,] 55.09181 1.630146
```

```
## Combine vectors by row
rbind(dat$weight, dat$height)
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
## [1,] 65.808579 63.452466 66.751126 75.953563 71.796896 59.965197 63.530817
## [2,]  1.608526  1.800125  1.694358  1.729665  1.420853  1.671726  1.623649
##           [,8]      [,9]    [,10]     [,11]     [,12]     [,13]     [,14]
## [1,] 83.475228 81.311799 59.23853 60.066263 69.151808 82.299773 54.982037
## [2,]  1.631499  1.736707  1.71830  1.656815  1.671839  1.833911  1.609441
##          [,15]     [,16]     [,17]     [,18]     [,19]     [,20]
## [1,] 70.533072 83.532770 61.614731 61.944175 65.034555 55.091813
## [2,]  1.859492  1.733384  1.612291  1.804116  1.629813  1.630146
```

Create your own functions

```
g <- function(x=34) {
  x + 100
}



# calculate the standardized values of a covariate
f <- function(x){
  sds <- (x - mean(x))/sd(x)
  return(sds)
}
f(12)
```

```
## [1] NA
```

```
f <- function(y) {
  (y - mean(y))/sd(y)
```

```
}
f(dat$weight)
```

```
##  [1] -0.2131188 -0.4682436 -0.1110578  0.8854013  0.4353088 -0.8458523
##  [7] -0.4597595  1.6998629  1.4656023 -0.9245373 -0.8349087  0.1488930
## [13]  1.5725822 -1.3854393  0.2984592  1.7060937 -0.6672374 -0.6315645
## [19] -0.2969317 -1.3735526
```

```
f(dat$height)
```

```
##  [1] -0.7556877  1.1669651  0.1056161  0.4599120 -2.6389432 -0.1214884
##  [7] -0.6039287 -0.5251569  0.5305858  0.3458685 -0.2711171 -0.1203580
## [13]  1.5060012 -0.7465040  1.7627011  0.4972317 -0.7179073  1.2070160
## [19] -0.5420725 -0.5387337
```

```
m <- 1
s <- 0.1

f <- function(y) {
  (y - m)/s
}
s<-0.2
f(dat$height)
```

```
##  [1] 3.042629 4.000624 3.471789 3.648323 2.104265 3.358630 3.118246
##  [8] 3.157495 3.683537 3.591498 3.284075 3.359193 4.169555 3.047205
## [15] 4.297460 3.666918 3.061454 4.020580 3.149067 3.150730
```

```
f1 <- function(x) {
  k <- x + 2
  f2 <- function(y) {
    y + 2
  }
  f2(k)
}

f1(10)
```

```
## [1] 14
```

```
#f2(10) # not working


f1 <- function(x) {
  m <- mean(x)
  s <- sd(x)
  function(x){(x-m)/s}
}
f2 <- f1(dat$height)
f(10)
```

```
## [1] 45
```

```
y<- 1:10
func4<-function(x){
  return(x+y )
}
```

```r
y<-6
func4(3)
```

```
## [1] 9
```

```r
# when a variable inside a function is not defined it will look for it in the
# environment where the function is defined
g<-function(x){
  y<-10
  func4(x)
}
```