

```

1   ###第3回欠測値処理.列or行の抽出.標準化得点の作成###
2   #最初にすること:dataを読み込み欠測値チェックをする
3   #欠測値があると計算しない関数がある(返値はNA)
4   #欠測値チェックと欠測値処理
5   s <- c("F", "F", "M", "M", "M")
6   h <- c(158, 162, 177, NA, 166) #4番目のデータが欠測
7   w <- c( 51, 55, 72, 57, 64)
8   df <- data.frame(s,h,w) #data.frame()関数:データフレーム化
9   anyNA(df) #anyNA()関数:返り値TRUEはNAあり
10  #欠測がある行(サンプル)を抽出,欠測内容をチェック
11  subset(df,
12  #subset()関数:条件式に合う行のみを抽出
13  complete.cases(df)==F)
14  #引数にcomplete.cases()関数:NA行を返す
15
16  ###欠測値処理(リストワイズ削除が基本)###
17  df.omit <- na.omit(df)
18  #na.omit()関数:NAをomitしたdata作成
19  #df.omit <- df[complete.cases(df),] #na.omit()関数と同じ
20  nrow(df.omit) #リストワイズ削除後のサンプル数
21  #リストワイズ削除したデータを保存
22  write.csv(df.omit, #write.csv()関数:保存data
23  "data/df.omit.csv") #保存場所と保存data名
24
25  ###重要:data入力時欠測値をNAと入力していれば問題なし
26  #blank:読み込み時NAになる.ピリオドや記号,数値で入力されている場合は
27  #NA指定する必要がある.指定しないとNA扱いにならない,
28  #data読込時,引数na.strings=()で指定する.
29  #以下はdata読込時の例である.
30  #na <- read.csv("data/NA.csv", #read.csv関数:data読込
31  # row.names = 1, #1列目ID:dataではない
32  # header = T, #1行目項目名:dataではない
33  # na.strings = (c("99", ".")) # "99"と"."をNAに指定
34
35  ###欠測値処理の実際###第2回Rの基本で用意したdataを使用
36  dat <- read.csv("data/cars93.csv", #derataフォルダのcars93
37  row.names = 1, #1列目ID:dataではない
38  header = T) #1行目項目名:dataではない
39  anyNA(dat) #返り値TRUE(NAあり)
40  na.dat <- subset(dat,
41  complete.cases(dat)==F) #NAの内容確認
42  #NAはボディタイプにより存在しないスペック(いわゆる欠測値ではない)
43  head(na.dat, 3) #最初から3行目まで表示
44
45  ###データから特定の列・行を抽出する###
46  ##特定の列を抽出:カンマ位置は前
47  dr <- dat[,c(5, 7, 12, 13)] #5, 7, 12, 13列の抽出
48  head(dr, 3)
49  ##特定の行を抽出:カンマ位置は後
50  dc <- dat[c(1:4),] #1行から4行
51  head(dc, 4)
52  ##値が特定の行を抽出:引数data名[data名$項目名,条件式]
53  #条件式:== 等号,!= 否定等号,>= 以上,>より大,<=以下,<より小
54  ds <- dat[dat$Type == "Small",] #TypeがSmallの行を抽出
55  table(ds$Type) #Smallが抽出されているか確認
56  dp <- dat[dat$Price <= 20,] #Priceが20以下の行を抽出
57  range(dp$Price) #Priceレンジが20以下か確認
58  length(dp$Price) #length()関数:サンプル数を確認
59  ##dataを標準化する##
60  z.dat <- scale(dat[,c(5, 7)]) #scale()関数:5, 7列を標準化

```