Poorman_Test : Miguel Gutierrez

Point 1

1) Add new columns to the following dataset which equate to the mean, median and the sum of each row.

```
df <- data.frame(x = runif(6), y = runif(6), z = runif(6))
df$Mean = apply(df[,c('x','y','z')],1,function(x) mean(x))
df$Median = apply(df[,c('x','y','z')],1,function(x) median(x))
df$Sum = apply(df[,c('x','y','z')],1,function(x) sum(x))
df
```

```
##             x         y         z      Mean    Median       Sum
## 1 0.102143964 0.1733999 0.7262316 0.3339252 0.1733999 1.0017755
## 2 0.171045222 0.6420551 0.8598410 0.5576471 0.6420551 1.6729413
## 3 0.114371173 0.6312633 0.7368168 0.4941504 0.6312633 1.4824513
## 4 0.006387657 0.1673915 0.2968683 0.1568825 0.1673915 0.4706475
## 5 0.104357530 0.6133544 0.9730602 0.5635907 0.6133544 1.6907722
## 6 0.301058741 0.8243485 0.4417500 0.5223857 0.4417500 1.5671572
```

Point 2

2) Using `stats::reshape()`, change the following dataset to wide format such that there is one row per sample and then reshape it back to the original shape.

```
df <- data.frame(sample = c(rep(1, 10), rep(2, 10)), test = rep(1:10, 2), values = runif(20))
df
```

```
##    sample test     values
## 1       1    1 0.26523494
## 2       1    2 0.65407520
## 3       1    3 0.46338474
## 4       1    4 0.30764912
## 5       1    5 0.82244981
## 6       1    6 0.63577023
## 7       1    7 0.09426367
## 8       1    8 0.17336880
## 9       1    9 0.79115033
## 10      1   10 0.88718959
## 11      2    1 0.27537904
## 12      2    2 0.71948132
## 13      2    3 0.67780993
## 14      2    4 0.75246942
## 15      2    5 0.74544726
```

```
## 16      2    6 0.62189455
## 17      2    7 0.90029442
## 18      2    8 0.87119785
## 19      2    9 0.98787353
## 20      2   10 0.58232147
```

we change the format such that there is one row per sample

```
reshaped = reshape(df, v.names = "values", idvar = "sample",timevar = "test", direction = "wide")
reshaped
```

```
##    sample  values.1  values.2  values.3  values.4  values.5  values.6
## 1       1 0.2652349 0.6540752 0.4633847 0.3076491 0.8224498 0.6357702
## 11      2 0.2753790 0.7194813 0.6778099 0.7524694 0.7454473 0.6218945
##      values.7  values.8  values.9 values.10
## 1  0.09426367 0.1733688 0.7911503 0.8871896
## 11 0.90029442 0.8711979 0.9878735 0.5823215
```

Now we are reshaping it back

```
original = reshape(reshaped, direction = "long")
original = original[order(original$sample),]
original
```

```
##      sample test     values
## 1.1       1    1 0.26523494
## 1.2       1    2 0.65407520
## 1.3       1    3 0.46338474
## 1.4       1    4 0.30764912
## 1.5       1    5 0.82244981
## 1.6       1    6 0.63577023
## 1.7       1    7 0.09426367
## 1.8       1    8 0.17336880
## 1.9       1    9 0.79115033
## 1.10      1   10 0.88718959
## 2.1       2    1 0.27537904
## 2.2       2    2 0.71948132
## 2.3       2    3 0.67780993
## 2.4       2    4 0.75246942
## 2.5       2    5 0.74544726
## 2.6       2    6 0.62189455
## 2.7       2    7 0.90029442
## 2.8       2    8 0.87119785
## 2.9       2    9 0.98787353
## 2.10      2   10 0.58232147
```

Point 3

3) Using `stats::reshape()`, reshape the following data to long format such that there are five columns: `sex`, `age`, `id`, `exam` and `score`. Then reshape the data back into the original format.

```r
df <- structure(list(
  sex = c(1L, 1L, 0L, 0L, 1L, 1L, 0L, 1L, 0L, 1L),
  age = rep(c(15L, 16L), 5),
  exam1 = c(34L, 47L, 41L, 44L, 47L, 42L, 57L, 61L, 53L, 42L),
  exam2 = c(46L, 54L, 47L, 41L, 65L, 41L, 62L, 59L, 61L, 39L),
  exam3 = c(45L, 49L, 40L, 40L, 60L, 57L, 63L, 49L, 61L, 42L),
  exam4 = c(39L, 53L, 39L, 50L, 50L, 72L, 55L, 44L, 57L, 42L),
  exam5 = c(36L, 61L, 51L, 26L, 56L, 31L, 41L, 66L, 56L, 41L)
), class = "data.frame", row.names = c(NA, -10L))
df
```

```
##    sex age exam1 exam2 exam3 exam4 exam5
## 1    1  15    34    46    45    39    36
## 2    1  16    47    54    49    53    61
## 3    0  15    41    47    40    39    51
## 4    0  16    44    41    40    50    26
## 5    1  15    47    65    60    50    56
## 6    1  16    42    41    57    72    31
## 7    0  15    57    62    63    55    41
## 8    1  16    61    59    49    44    66
## 9    0  15    53    61    61    57    56
## 10   1  16    42    39    42    42    41
```

First we reshape in long format

```r
reshaped = reshape(df, sep ="",varying = 3:7, direction = "long")
reshaped = reshaped[order(reshaped$sex,reshaped$age),]
names(reshaped)[names(reshaped)=="exam"] = "score"
names(reshaped)[names(reshaped)=="time"] = "exam"
reshaped
```

```
##      sex age exam score id
## 3.1    0  15    1    41  3
## 7.1    0  15    1    57  7
## 9.1    0  15    1    53  9
## 3.2    0  15    2    47  3
## 7.2    0  15    2    62  7
## 9.2    0  15    2    61  9
## 3.3    0  15    3    40  3
## 7.3    0  15    3    63  7
## 9.3    0  15    3    61  9
## 3.4    0  15    4    39  3
## 7.4    0  15    4    55  7
## 9.4    0  15    4    57  9
## 3.5    0  15    5    51  3
## 7.5    0  15    5    41  7
## 9.5    0  15    5    56  9
## 4.1    0  16    1    44  4
## 4.2    0  16    2    41  4
## 4.3    0  16    3    40  4
## 4.4    0  16    4    50  4
## 4.5    0  16    5    26  4
```

```
## 1.1    1  15    1    34  1
## 5.1    1  15    1    47  5
## 1.2    1  15    2    46  1
## 5.2    1  15    2    65  5
## 1.3    1  15    3    45  1
## 5.3    1  15    3    60  5
## 1.4    1  15    4    39  1
## 5.4    1  15    4    50  5
## 1.5    1  15    5    36  1
## 5.5    1  15    5    56  5
## 2.1    1  16    1    47  2
## 6.1    1  16    1    42  6
## 8.1    1  16    1    61  8
## 10.1   1  16    1    42 10
## 2.2    1  16    2    54  2
## 6.2    1  16    2    41  6
## 8.2    1  16    2    59  8
## 10.2   1  16    2    39 10
## 2.3    1  16    3    49  2
## 6.3    1  16    3    57  6
## 8.3    1  16    3    49  8
## 10.3   1  16    3    42 10
## 2.4    1  16    4    53  2
## 6.4    1  16    4    72  6
## 8.4    1  16    4    44  8
## 10.4   1  16    4    42 10
## 2.5    1  16    5    61  2
## 6.5    1  16    5    31  6
## 8.5    1  16    5    66  8
## 10.5   1  16    5    41 10
```

then again in original format

```
reshape(reshaped, v.names = "score",timevar = "exam",direction = "wide")
```

```
##      sex age id score.1 score.2 score.3 score.4 score.5
## 3.1    0  15  3      41      47      40      39      51
## 7.1    0  15  7      57      62      63      55      41
## 9.1    0  15  9      53      61      61      57      56
## 4.1    0  16  4      44      41      40      50      26
## 1.1    1  15  1      34      46      45      39      36
## 5.1    1  15  5      47      65      60      50      56
## 2.1    1  16  2      47      54      49      53      61
## 6.1    1  16  6      42      41      57      72      31
## 8.1    1  16  8      61      59      49      44      66
## 10.1   1  16 10      42      39      42      42      41
```

Point 4

4) Using `stats::reshape()`, reshape the following data to a wide format with the resulting columns: `id`, `min.1`, `max.1`, `min.2`, `max.2`. Then reshape the data back to the original shape.

```r
df <- data.frame(id = rep(1:4, each = 2), sample = rep(c(1, 2), 4), min = 1:8, max = 3:10)
df
```

```
##   id sample min max
## 1  1      1   1   3
## 2  1      2   2   4
## 3  2      1   3   5
## 4  2      2   4   6
## 5  3      1   5   7
## 6  3      2   6   8
## 7  4      1   7   9
## 8  4      2   8  10
```

First we reshape the data as proposed in the testa

```r
reshaped = reshape(df, v.names = c("min","max"), idvar = "id",timevar = "sample", direction = "wide")
reshaped
```

```
##   id min.1 max.1 min.2 max.2
## 1  1     1     3     2     4
## 3  2     3     5     4     6
## 5  3     5     7     6     8
## 7  4     7     9     8    10
```

Now we reshaped the data back to the original shape

```r
original = reshape(reshaped, varying=2:5, direction = "long")
names(original)[names(original)=="time"] = "sample"
original
```

```
##     id sample min max
## 1.1  1      1   1   3
## 2.1  2      1   3   5
## 3.1  3      1   5   7
## 4.1  4      1   7   9
## 1.2  1      2   2   4
## 2.2  2      2   4   6
## 3.2  3      2   6   8
## 4.2  4      2   8  10
```