Problem Set 2 - Solutions

Applied Statistics and Econometrics II, Spring 2018

Ercan Karadas

February 15, 2018

Contents

Problem 1

```
# a)
x <- c(-20, -15, -5, 8, 12, 9, 2, 23, 19)
sum(x)
```

```
## [1] 33
```

```
# b)
mean(x)
```

```
## [1] 3.666667
```

```
# c)
sum(x)/length(x)
```

```
## [1] 3.666667
```

```
# d)
sum(x[x>0])
```

```
## [1] 73
```

```
# e)
(sum(x)-max(x))/(length(x)-1)
```

```
## [1] 1.25
```

```
# f)
x[abs(x)>=8 & x<8]
```

```
## [1] -20 -15
```

Problem 2

```
# a)
x <- c(1, 8, 2, 6, 3, 8, 5, 5, 5, 5)
sum(x)-x[3]-x[5]
```

```
## [1] 43
```

```
sum(x[c(1,2,4,6,7,8,9,10)])
```

```
## [1] 43
```

```
# b)
sum(x[1:6])
```

```
## [1] 28
```

```
sum(x[!x == 5] )
```

```
## [1] 28
```

```
# c)
replace(x, x==8, 7)
```

```
##  [1] 1 7 2 6 3 7 5 5 5 5
```

Problem 3

```
# a)
data <- rnorm(10*5, mean=5, sd=sqrt(2))
M <- matrix(data, 10, 5)
M
```

```
##           [,1]     [,2]     [,3]     [,4]     [,5]
##  [1,] 6.215446 3.465399 5.251192 7.258679 4.080469
##  [2,] 6.667296 4.857021 4.363544 4.533525 6.043499
##  [3,] 5.888550 3.883523 3.493054 4.126665 5.365629
##  [4,] 5.627349 7.027037 4.323208 4.246133 3.377965
##  [5,] 3.190189 2.260458 3.474537 7.781038 6.459272
##  [6,] 4.477239 5.850680 7.936552 3.845347 4.982081
##  [7,] 5.695725 3.803008 4.557818 6.102735 5.324515
##  [8,] 3.065166 6.343886 3.541402 5.876167 4.040029
##  [9,] 5.511673 5.481425 6.315896 4.785727 4.278635
## [10,] 3.382904 6.481627 6.105171 4.423752 7.520998
```

```
# b)
N <- matrix(0, 10, 5)
na_loc <- sample(1:50, 5, replace = FALSE)
N[na_loc] <- NA
N
```

```
##       [,1] [,2] [,3] [,4] [,5]
##  [1,]    0    0    0    0    0
##  [2,]    0    0    0    0    0
##  [3,]    0    0    0    0    0
##  [4,]   NA    0    0    0    0
##  [5,]    0    0    0    0    0
##  [6,]    0    0    0    0   NA
##  [7,]   NA    0    0   NA    0
##  [8,]    0    0    0    0    0
##  [9,]    0    0    0   NA    0
## [10,]    0    0    0    0    0
```

```
# c)
P <- M+N
P
```

```
##           [,1]     [,2]     [,3]     [,4]     [,5]
##  [1,] 6.215446 3.465399 5.251192 7.258679 4.080469
##  [2,] 6.667296 4.857021 4.363544 4.533525 6.043499
##  [3,] 5.888550 3.883523 3.493054 4.126665 5.365629
##  [4,]       NA 7.027037 4.323208 4.246133 3.377965
##  [5,] 3.190189 2.260458 3.474537 7.781038 6.459272
##  [6,] 4.477239 5.850680 7.936552 3.845347       NA
##  [7,]       NA 3.803008 4.557818       NA 5.324515
##  [8,] 3.065166 6.343886 3.541402 5.876167 4.040029
##  [9,] 5.511673 5.481425 6.315896       NA 4.278635
## [10,] 3.382904 6.481627 6.105171 4.423752 7.520998
```

```
# d)
is.na(P)
```

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

```
ToBeDeleted <- which(rowSums(is.na(P)) > 0)

P[-ToBeDeleted, ]
```

```
##          [,1]     [,2]     [,3]     [,4]     [,5]
## [1,] 6.215446 3.465399 5.251192 7.258679 4.080469
## [2,] 6.667296 4.857021 4.363544 4.533525 6.043499
## [3,] 5.888550 3.883523 3.493054 4.126665 5.365629
## [4,] 3.190189 2.260458 3.474537 7.781038 6.459272
## [5,] 3.065166 6.343886 3.541402 5.876167 4.040029
## [6,] 3.382904 6.481627 6.105171 4.423752 7.520998
```

Problem 4

```
data("chickwts")
mean(chickwts$weight[chickwts$feed == "horsebean"])
```

```
## [1] 160.2
```

Problem 5

```
# a)
my_vec <- c(1:100)
my_vec2 <- my_vec/3
my_vec2
```

```
##   [1]  0.3333333  0.6666667  1.0000000  1.3333333  1.6666667  2.0000000
##   [7]  2.3333333  2.6666667  3.0000000  3.3333333  3.6666667  4.0000000
##  [13]  4.3333333  4.6666667  5.0000000  5.3333333  5.6666667  6.0000000
##  [19]  6.3333333  6.6666667  7.0000000  7.3333333  7.6666667  8.0000000
##  [25]  8.3333333  8.6666667  9.0000000  9.3333333  9.6666667 10.0000000
##  [31] 10.3333333 10.6666667 11.0000000 11.3333333 11.6666667 12.0000000
##  [37] 12.3333333 12.6666667 13.0000000 13.3333333 13.6666667 14.0000000
##  [43] 14.3333333 14.6666667 15.0000000 15.3333333 15.6666667 16.0000000
##  [49] 16.3333333 16.6666667 17.0000000 17.3333333 17.6666667 18.0000000
##  [55] 18.3333333 18.6666667 19.0000000 19.3333333 19.6666667 20.0000000
##  [61] 20.3333333 20.6666667 21.0000000 21.3333333 21.6666667 22.0000000
##  [67] 22.3333333 22.6666667 23.0000000 23.3333333 23.6666667 24.0000000
##  [73] 24.3333333 24.6666667 25.0000000 25.3333333 25.6666667 26.0000000
```

```
##  [79] 26.3333333 26.6666667 27.0000000 27.3333333 27.6666667 28.0000000
##  [85] 28.3333333 28.6666667 29.0000000 29.3333333 29.6666667 30.0000000
##  [91] 30.3333333 30.6666667 31.0000000 31.3333333 31.6666667 32.0000000
##  [97] 32.3333333 32.6666667 33.0000000 33.3333333
```

```
# b)
sum(my_vec)/length(my_vec)
```

```
## [1] 50.5
```

```
# c)
my_vec3 <- my_vec[20:35]
my_vec3
```

```
##  [1] 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35
```

Problem 5

```
mean(rnorm(1000))
```

```
## [1] -0.02490296
```

The output is the mean of the 1,000 random values on generated on a normal distribution with mean zero and unit variance. Because of the LLN, this value would be very close to 0.

Problem 7

```
x1 <- rnorm(10)
x2 <- rnorm(10)
T <- data.frame("a" = x1, "b" = x1 + x2)
T
```

```
##              a          b
## 1  -1.04834068  0.2478602
## 2   1.20926385  1.6088382
## 3  -0.30902862  0.1556083
## 4  -0.24337977  0.1407918
## 5  -0.20654939 -1.5297613
## 6  -0.08181453 -0.7124707
## 7  -1.08536836 -1.3377046
## 8   0.45682448  1.4409454
## 9   0.96432991  4.3722557
## 10  0.41037429  1.0142425
```

Problem 8

```
orange <- head(Orange, n=2)
orange
```

```
##   Tree age circumference
## 1    1 118            30
```

```
## 2    1 484            58
```

```
Y <- orange["circumference"]
X <- sqrt(orange["age"])
Y + X
```

```
##   circumference
## 1      40.86278
## 2      80.00000
```

Problem 9

```
orange <- head(Orange)
orange$age[orange$age <= 900] <- 0
orange
```

```
##   Tree  age circumference
## 1    1    0            30
## 2    1    0            58
## 3    1    0            87
## 4    1 1004           115
## 5    1 1231           120
## 6    1 1372           142
```

Problem 10

```
P <- seq(30,90,2)
Q <- matrix(data = P, nrow = 6, ncol = 5)
```

```
## Warning in matrix(data = P, nrow = 6, ncol = 5): data length [31] is not a
## sub-multiple or multiple of the number of rows [6]
```

```
Q
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]   30   42   54   66   78
## [2,]   32   44   56   68   80
## [3,]   34   46   58   70   82
## [4,]   36   48   60   72   84
## [5,]   38   50   62   74   86
## [6,]   40   52   64   76   88
```

Problem 11

```
my_function <- function(x){
  x <- input1
  return(sum(x))
  return(mean(x))
  if(integer(String)) stop('Data is not numeric!')
  }
```

Problem 12

```
x <- rep(1:40)
sum(x)
```

```
## [1] 820
```

Problem 13

```
# a)
mean(ChickWeight[,1])
```

```
## [1] 121.8183
```

```
mean(ChickWeight[,3])
```

```
## Warning in mean.default(ChickWeight[, 3]): argument is not numeric or
## logical: returning NA
```

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

```
# b)
mean(as.numeric(ChickWeight[ ,3]))
```

```
## [1] 26.25952
```

Problem 14

```
scores <- c(73, 74, 92, 98, 100, 72, 74, 85, 76, 94, 89, 73, 76, 99)
mean(scores)
```

```
## [1] 83.92857
```

```
mean(scores, trim=.2)
```

```
## [1] 83.1
```

```
median(scores)
```

```
## [1] 80.5
```

Problem 15

```
sample <- c(20, 121, 132, 123, 145, 151, 119, 133, 134, 130, 200)
standev <- sd(sample)
mean <- mean(sample)
arg1 <- sample > (mean + 2*standev)
arg2 <- sample < (mean - 2*standev)
xi <- sample[arg1:arg2]
```

```
## Warning in arg1:arg2: numerical expression has 11 elements: only the first
## used
```

```
## Warning in arg1:arg2: numerical expression has 11 elements: only the first
## used
```

```
xi
```

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

Problem 16

```
# b)
read.table("lunatics.txt")
```

```
##            V1  V2   V3      V4   V5    V6
## 1      COUNTY NBR DIST     POP PDEN PHOME
## 2   BERKSHIRE 119   97  26.656   56    77
## 3    FRANKLIN  84   62  22.260   45    81
## 4   HAMPSHIRE  94   54  23.312   72    75
## 5     HAMPDEN 105   52  18.900   94    69
## 6   WORCESTER 351   20  82.836   98    64
## 7   MIDDLESEX 357   14  66.759  231    47
## 8       ESSEX 377   10  95.004 3252    47
## 9     SUFFOLK 458    4 123.202 3042     6
## 10    NORFOLK 241   14  62.901  235    49
## 11    BRISTOL 158   14  29.704  151    60
## 12   PLYMOUTH 139   16  32.526   91    68
## 13 BARNSTABLE  78   44  16.692   93    76
## 14  NANTUCKET  12   77   1.740  179    25
## 15      DUKES  19   52   7.524   46    79
```

```
# c)
lunatics <- read.table("lunatics.txt")
str(lunatics)
```

```
## 'data.frame':    15 obs. of  6 variables:
##  $ V1: Factor w/ 15 levels "BARNSTABLE","BERKSHIRE",..: 4 2 7 9 8 15 10 6 14 12 ...
##  $ V2: Factor w/ 15 levels "105","119","12",..: 15 2 13 14 1 8 9 10 11 7 ...
##  $ V3: Factor w/ 12 levels "10","14","16",..: 12 11 9 8 7 4 2 1 5 2 ...
##  $ V4: Factor w/ 15 levels "1.740","123.202",..: 15 7 5 6 4 13 11 14 2 10 ...
##  $ V5: Factor w/ 15 levels "151","179","231",..: 15 9 7 10 13 14 3 6 5 4 ...
##  $ V6: Factor w/ 14 levels "25","47","49",..: 14 11 13 9 8 6 2 2 4 3 ...
```

```
# d)
write.csv("R Practice", "lunatics.csv")

# e)
library("readr")
read_csv("lunatics.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   X1 = col_double(),
##   x = col_character()
```

```
## )
```

```
## # A tibble: 1 x 2
##      X1 x
##   <dbl> <chr>
## 1     1 R Practice
```

Problem 17

```
# a)
PractData <- read.table("PiDigits.dat.txt", skip=60, stringsAsFactors = FALSE, sep = ",")
write.csv("R Practice", '17aData.csv')

# b)
PractDataB <- read.csv("PiDigits.dat.txt", stringsAsFactors = FALSE, skip=110, sep = ",")
write.csv("R Practice", 'PiDigitsSkipped,csv')

# c)
head(PractData, n = 100)

# d)
tablePractData <- table(PractData)
tablePractData

# e)
barplot(tablePractData)
```

Problem 18

```
mydata <- data.frame(x1 = c(2, 2, 6, 4), x2 = c(3, 4, 2, 8))
mydata$sumx <- mydata$x1 + mydata$x2
mydata$meanx <- (mydata$x1 + mydata$x2)/2
mydata
```

```
##   x1 x2 sumx meanx
## 1  2  3    5   2.5
## 2  2  4    6   3.0
## 3  6  2    8   4.0
## 4  4  8   12   6.0
```

Problem 19

```
manager <- c(1, 2, 3, 4, 5)
date <- c("10/24/08" , "10/28/08" , "10/1/08" , "10/12/08" , "5/1/09")
country <- c("US", "US", "UK", "UK", "UK")
gender <- c("M", "F", "F", "M", "F")

age <- c(32, 45, 25, 39, 99)
```

```
q1 <- c(5, 3, 3, 3, 2)
q2 <- c(4, 5, 5, 3, 2)
q3 <- c(5, 2, 5, 4, 1)
q4 <- c(5, 5, 5, NA, 2)
q5 <- c(5, 5, 2, NA, 1)

leadership <- data.frame(manager, date, country, gender, age,
q1 , q2 , q3 , q4 , q5 , stringsAsFactors=FALSE)

# a)
leadership$age[leadership$age == 99] <- NA

# b)
leadership$agecat[leadership$age > 75] <- "Elder"
leadership$agecat[leadership$age >= 55 & leadership$age <= 75] <- "Middle Aged"
leadership$agecat[leadership$age < 55] <- "Young"

# c)
ordered(age)
```

```
## [1] 32 45 25 39 99
## Levels: 25 < 32 < 39 < 45 < 99
```

```
# d)
names(leadership)[1] <- "managerID"
names(leadership)[5] <- "ages"

# e)
is.na(leadership[4:5, 6:10])
```

```
##      q1    q2    q3    q4    q5
## 4 FALSE FALSE FALSE  TRUE  TRUE
## 5 FALSE FALSE FALSE FALSE FALSE
```

```
# f)
myformat <- "%m/%d/%y"
leadership$date <- as.Date(leadership$date , myformat)
```

Problem 20

```
sum(c(1, -2, NA, 3))
```

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

```
sum(c(1, -2, NA, 3), na.rm=TRUE)
```

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

The second function removes missing values from the function, thus allowing the sum function to run correctly

Problem 21

```
leadership
```

```
##   managerID       date country gender ages q1 q2 q3 q4 q5 agecat
## 1         1 2008-10-24      US      M   32  5  4  5  5  5  Young
## 2         2 2008-10-28      US      F   45  3  5  2  5  5  Young
## 3         3 2008-10-01      UK      F   25  3  5  5  5  2  Young
## 4         4 2008-10-12      UK      M   39  3  3  4 NA NA  Young
## 5         5 2009-05-01      UK      F   NA  2  2  1  2  1   <NA>
```

```
leadership_wo_NA <- na.omit(leadership)
leadership_wo_NA
```

```
##   managerID       date country gender ages q1 q2 q3 q4 q5 agecat
## 1         1 2008-10-24      US      M   32  5  4  5  5  5  Young
## 2         2 2008-10-28      US      F   45  3  5  2  5  5  Young
## 3         3 2008-10-01      UK      F   25  3  5  5  5  2  Young
```

Problem 22

```
vector22 <- c(1:100)
vector22[seq(3, length(vector22), 3)] <- NA
vector22
```

```
##   [1]   1   2  NA   4   5  NA   7   8  NA  10  11  NA  13  14  NA  16  17
##  [18]  NA  19  20  NA  22  23  NA  25  26  NA  28  29  NA  31  32  NA  34
##  [35]  35  NA  37  38  NA  40  41  NA  43  44  NA  46  47  NA  49  50  NA
##  [52]  52  53  NA  55  56  NA  58  59  NA  61  62  NA  64  65  NA  67  68
##  [69]  NA  70  71  NA  73  74  NA  76  77  NA  79  80  NA  82  83  NA  85
##  [86]  86  NA  88  89  NA  91  92  NA  94  95  NA  97  98  NA 100
```

```
sum(is.na(vector22))
```

```
## [1] 33
```

Problem 23

```
# a)
leadershipOriginal <- leadership
leadershipOriginal
```

```
##   managerID       date country gender ages q1 q2 q3 q4 q5 agecat
## 1         1 2008-10-24      US      M   32  5  4  5  5  5  Young
## 2         2 2008-10-28      US      F   45  3  5  2  5  5  Young
## 3         3 2008-10-01      UK      F   25  3  5  5  5  2  Young
## 4         4 2008-10-12      UK      M   39  3  3  4 NA NA  Young
## 5         5 2009-05-01      UK      F   NA  2  2  1  2  1   <NA>
```

```
newdata <- leadership[order(leadership$ages),]
newdata
```

```
##   managerID       date country gender ages q1 q2 q3 q4 q5 agecat
## 3         3 2008-10-01      UK      F   25  3  5  5  5  2  Young
## 1         1 2008-10-24      US      M   32  5  4  5  5  5  Young
## 4         4 2008-10-12      UK      M   39  3  3  4 NA NA  Young
## 2         2 2008-10-28      US      F   45  3  5  2  5  5  Young
## 5         5 2009-05-01      UK      F   NA  2  2  1  2  1   <NA>
```

```
# b)
newdata <- leadership[order(leadership$gender, leadership$ages),]
newdata
```

```
##   managerID       date country gender ages q1 q2 q3 q4 q5 agecat
## 3         3 2008-10-01      UK      F   25  3  5  5  5  2  Young
## 2         2 2008-10-28      US      F   45  3  5  2  5  5  Young
## 5         5 2009-05-01      UK      F   NA  2  2  1  2  1   <NA>
## 1         1 2008-10-24      US      M   32  5  4  5  5  5  Young
## 4         4 2008-10-12      UK      M   39  3  3  4 NA NA  Young
```

```
# c)
newdata <- leadership[order(leadership$gender, -leadership$ages),]
newdata
```

```
##   managerID       date country gender ages q1 q2 q3 q4 q5 agecat
## 2         2 2008-10-28      US      F   45  3  5  2  5  5  Young
## 3         3 2008-10-01      UK      F   25  3  5  5  5  2  Young
## 5         5 2009-05-01      UK      F   NA  2  2  1  2  1   <NA>
## 4         4 2008-10-12      UK      M   39  3  3  4 NA NA  Young
## 1         1 2008-10-24      US      M   32  5  4  5  5  5  Young
```