Week-3-assignment.R

Admin

Thu Sep 11 18:29:10 2014

```
# Functions implemented to answer the questions in the quiz

# myPercentile
#Function that accepts a numerical vector x and the percentile p and then
returns the numerical number
# that corresponds the the percentile or the probability is a fraction
#Reference Quartiles formula to use: Journet, David (March 1999). How to
calculate them? Retrieved on September 10th, 2014
#from http://www.haiweb.org/medicineprices/manual/quartiles_iTSS.pdf


myPercentile = function (x, p){
# first sort the vector
x = sort(x)
n = length(x)

# this is the formula if we want to use the SAS model
# g = ((n+1)*p) %% (as.integer((n+1)*p))
# j = as.integer((n+1)*p)
# y = (1-g) * x[j] + g * (x[j+1])

# Excel which is the same as in R
    if ((n-1)*p <1){
        g = ((n-1)*p) %% 1
    } else
        {g = ((n-1)*p) %% (as.integer((n-1)*p))
    }

j = as.integer((n-1)*p)

y = (1-g) * x[j+1] + g * (x[j+2])

return (y)

}

# 1. Write a function that takes a vector as input and returns the number of
missing values in the vector.

missingValues = function (x){
    return (length(x) - length(na.omit(x)))
```

```
  }


# 2. Write a function that takes a data frame as input and returns a named
vector with the number of missing
#    values in each column of the data frame. (The names of the entries
should be the corresponding column names
#    of the data frame.) You may use the function from the previous question
as part of your solution.

missingValuesDframe = function (df) {
    return (sapply(df, missingValues))
}


# 3. Write a function that takes a numeric vector as input and uses it to
determine the minimum, the maximum,
#    the mean, the median, the first quartile, the third quartile, the
standard deviation of the vector, and the
#    number of missing values. Do not use any built-in functions to do this.
Return a named list with the eight
#    desired values in an order you deem best. (You may, if you like, use the
function you wrote for question 1.)

# x <- letters[1:2]
# y <- 1:2
# x
# y
# mapply(function(x,y) y,x,y,SIMPLIFY = TRUE,USE.NAMES = TRUE)

myStats = function(v){


 #first count the number of missing values and assign them to a variable
missingValues
    mValues = missingValues(v)

 # take out the missing values
    v =   v[!is.na(v)]

 #then calculate the mean
    mMean = (sum(v)/length(v))

 #calculate the minimum
 v
 mMin = v[1]
    for (i in 1:length(v - 1)){
        mMin = ifelse((mMin > v[i]), v[i], mMin)
        #print (paste(i, v[i], mMin))
```

```
    }
 # calculate the maximum

 mMax = v[1]
    for (i in 1:length(v - 1)){
        mMax = ifelse((mMax > v[i]), mMax, v[i])
        #print (paste(i, v[i], mMax))
    }


# Calcualte standard deviation
#ref for code: http://stats.stackexchange.com/questions/25956/what-formula-
is-used-for-standard-deviation-in-r

    mSD = sqrt((sum((v - mMean)^2) / (length(v) - 1)))

# calculate the median percentile = 0.5
    mMedian = myPercentile(v, 0.5)

# calculate the first Quartile,  percentile = 0.25
    mQ1 = myPercentile(v, 0.25)

# calculate the thrid Quartile,  percentile = 0.75
    mQ3 = myPercentile(v, 0.75)

# return named list

statResults = list(Minimum = mMin, Q1 = mQ1, Median = mMedian, Mean = mMean,
Q3 = mQ3, Maximum = mMax, SD = mSD, Missing_Values = mValues)

return(statResults)

}


# 4. Write a function that takes a character or factor vector and determines
the number of distinct elements in
#    the vector, the most commonly occurring element, the number of times the
most commonly occurring element
#    occurs, and the number of missing values. (Be sure to handle ties
gracefully.) Have the function return a
#    named list with the desired information in a logical order.

myDistinctFun = function (s){

    missingValues = length (s) - length(na.omit(s))
    #take out the missing values
    s =   s[!is.na(s)]
```

```
    # convert s to a factor f
    f = as.factor(s)
    levels(f)
    #number of levels
    nlevels(f)
    f
    t = table(f)
    t
    max(t)
    ls = list()
    for (i in 1:(nlevels(f))){

        if (t[[i]] == max(t))
        {
            ls = append(ls, names(t)[i])
        }
    }
    ls

    nComment = ''
    if (length(ls) > 1) {

        nComment = paste(nComment, "Number of elements with  a tie are: ",
length(ls))

        for (i in 1:length(ls)){
            nComment = paste(nComment, ", ", ls[[i]] )
        }
    }

    li = list(Num_Levels = nlevels(f), Most_Common = ls[[1]], Num_Most_Common
= max(t), Num_Missing = missingValues, Comments = nComment  )
    return (li)
}



# 5. Write a function that takes a logical vector and determines the number
of true values, the number of false
#    values, the proportion of true values, and the number of missing values.
Have the function return a named list
#    with the desired information in a logical order.

myLogicFun = function (v){
numMissing = length (v) - length(na.omit(v))
numNotMissing = length(!na.omit(v))
#take out the missing values
v =   v[!is.na(v)]
countTrue = 0
```

```
for (i in 1:numNotMissing){

    if (v[i] == TRUE) {
        #print (v[i])
        countTrue = countTrue + 1
    }
}

myList = list(True = countTrue, False = (numNotMissing - countTrue), P_True =
(countTrue/numNotMissing), Missing_Values = numMissing)
return (myList)
}


# 6. Write a function that takes as its input a data frame and returns a
summary of its columns using the functions
#    you write for questions 3-5. You may assume that all columns will be of
the three types in those questions.
#     You are expected to use the functions you have written in the previous
questions, so you do not have to write
#     them again by scratch. Return the desired information in a format that
you deem best.
#(One suggestion would be a named list of lists, but I leave it to your
judgment.)



funQ6 = function(w) {

ls = list(Statistics = myStats(w[,1]), Character_Vector =
myDistinctFun(w[,2]), Logical_Vector = myLogicFun(w[,3]))

return (ls)
}

##############################################################################
########################################
#**************************** Code to test the
function**************************************************
#
##############################################################################
########################################
s = read.csv("C:/Users/Admin/Desktop/Q6.csv", header = TRUE, sep = ",",
stringsAsFactors = FALSE)
#add some NA's to our vector for the first column
n = sample(1:100, 33)
s[n,1] = NA

#add some NA's to our vector for the second column
```

```r
n = sample(1:100, 19)
s[n,2] = NA

result = funQ6(s)
result
```

```
## $Statistics
## $Statistics$Minimum
## [1] 5
##
## $Statistics$Q1
## [1] 41
##
## $Statistics$Median
## [1] 55
##
## $Statistics$Mean
## [1] 58.85
##
## $Statistics$Q3
## [1] 87
##
## $Statistics$Maximum
## [1] 100
##
## $Statistics$SD
## [1] 26.77
##
## $Statistics$Missing_Values
## [1] 33
##
##
## $Character_Vector
## $Character_Vector$Num_Levels
## [1] 7
##
## $Character_Vector$Most_Common
## [1] "red"
##
## $Character_Vector$Num_Most_Common
## [1] 14
##
## $Character_Vector$Num_Missing
## [1] 19
##
## $Character_Vector$Comments
## [1] ""
##
##
## $Logical_Vector
```

```
## $Logical_Vector$True
## [1] 36
##
## $Logical_Vector$False
## [1] 29
##
## $Logical_Vector$P_True
## [1] 0.5538
##
## $Logical_Vector$Missing_Values
## [1] 35
```

```
#*************************************************************************
************************************
```

```
#*****************************Resuts*************************************
************************************
```

```
# result
# $Statistics
# $Statistics$Minimum
# [1] 5
#
# $Statistics$Q1
# [1] 34.5
#
# $Statistics$Median
# [1] 54
#
# $Statistics$Mean
# [1] 55.62687
#
# $Statistics$Q3
# [1] 75.5
#
# $Statistics$Maximum
# [1] 100
#
# $Statistics$SD
# [1] 26.48432
#
# $Statistics$Missing_Values
# [1] 33
#
#
# $Character_Vector
# $Character_Vector$Num_Levels
# [1] 7
#
# $Character_Vector$Most_Common
```

```
# [1] "green"
#
# $Character_Vector$Num_Most_Common
# [1] 15
#
# $Character_Vector$Num_Missing
# [1] 19
#
# $Character_Vector$Comments
# [1] ""
#
#
# $Logical_Vector
# $Logical_Vector$True
# [1] 36
#
# $Logical_Vector$False
# [1] 29
#
# $Logical_Vector$P_True
# [1] 0.5538462
#
# $Logical_Vector$Missing_Values
# [1] 35
```