Beginner Questions

1. What **class** of object is mtcars? What function did you use to find out?

The data set “mtcars” is a data.frame object class. I used the command > class(mtcars)

1. Is precip defined as a **1-dimensional array** or a **vector**? How did you find out?

> is(precip, “vector”)

Precip is a vector.

1. How would you convert the **data.frame** trees into a matrix?

> as(trees, “matrix”)

1. What is the name of the 14th city in the precip dataset?

Atlanta

1. What function would you use if you wanted to combine all three data sets into a single object?

Data.frame function

>MyFrame<-data.frame(precip,trees,mtcars)

1. Does precip consist of numeric data? How did you find out?

> is(precip, “numeric”)

Precip is numeric data.

1. Code **four** different ways to **subscript** the **2nd row** and **7th column** of mtcars using bracket notation - i.e., 17.02.

>Mtcars[2,7]

>Mtcars[“Mazda RX4 Wag”, “qsec”]

>Mtcars[2, “qsec”]

>Mtcars[ , “qsec”][2]

1. How would you change the precipitation values of "Juneau", "Phoenix", and "Sacramento" to 23, 46, and 12 in the precip dataset. (Hint: You will need to use **subscripts** and the <- operator).

>Precip[“Juneau”] <- 23

>Precip[“Phoenix”] <- 46

>Precip[“Sacramento”] <- 12

1. Are there **any** trees in the trees dataset with more **girth** than **volume**? How did you find out?

>Trees[“Girth”]>trees[“Volume”]

Gives an output of FALSE for all rows

1. Take the sum of all elements in column **height** of the trees dataset, call this value **A**. Take the sum of all elements in row **Valiant** of the **mtcars** dataset, call this value **B**. Take the sum of the first **8 elements** of the **precip** dataset, call this value **C**. Divide **C** by **B** and add **A**. What is your final answer?

>A <- sum(trees[“Height”])

#A= 2356

>B <- sum(mtcars[“Valiant”, ]

#B=385.54

>C <- sum(precip[1:8])

#C= 249.4

>(C/B)+A

# 2356.647

Intermediate Questions

1. What does the REPLACE= argument of the sample( ) function do?

This allows for the matrix to fill in the sample size that is larger than the population size (i.e. there are only two elements, true and false, so you need to say replace=true so that it repeats)

1. Using as(MyMatrix,"numeric") will not convert MyMatrix to numeric data! Can you think of a property of logicals that you can use to convert the logicals to 0's and 1's other than the as( ) function?

>MyMatrix+0 converts the matrix into numeric data since R thinks the logical data is numeric.

1. If you wanted to check if **all** of the elements in each row are true, how would you do this?

>Is(MyMatrix[1], “TRUE”)

#Repeated for each desired row

#The which() function can tell you which elements in the row are true

1. How many times does the number 7 occur in MyMatrix?

>Which(MyMatrix==7)

>“seven” <- which(MyMatrix==7)

#The number 7 occurs 16 times

1. How do you find the sum of each column?

>Apply(MyMatrix,2,sum)

1. How do you find the product of each column?

>Apply(MyMatrix,2,prod)

1. How would you change every instance of the number 10 to 12?

>MyMatrix[which(MyMatrix==10)]<-12

#this converts each instance of a 10 into a 12

1. How many values in MyMatrix are greater than 3 and less than 8?

>Which (MyMatrix>3 & MyMatrix<8)

>“numbereight” <- which(MyMatrix>3 & Matrix<8)

>length(numbereight)

#There are 33 elements that are greater than 3 and less than 8

1. How do you change the elements of column 12 into **character data**, while keeping columns 1- 11 as numeric data??

>MyMatrix<-as.data.frame(MyMatrix)

>MyMatrix[,12]<- as.character(MyMatrix[,12])

#The matrix needs to be converted into a data frame first so you can make column 12 character data

1. Find which rows of MyMatrix have a sum >70. Make a new version of MyMatrix where the 13th column is a set of TRUE and FALSE values denoting which rows have a sum >70.

#(this is in the original MyMatrix without changing each value of 10 to 12)

>”seventy”<-apply(MyMatrix,1,sum)

>which(seventy>70)

#rows 2,5,6,7, and 8 all sum to >70

#second part of the question

>seventy<- apply(MyMatrix,1,sum)>70

#gives a vector of true/false values

>MyFrame<-data.frame(MyMatrix,seventy)

>MyFrame #gives a data frame of the matrix with an additional column of true/false

Advanced Questions

1. Write a function that returns the phrase "Hello, World."

>numberone<-function(Argument){print(Argument)}

>numberone(“Hello, World”)

1. Load the iris dataset we used in the earlier tests. Write a function that takes iris as its argument, and returns three subsets of the data.frame split by the three different types of species (saved as a single object).

> Irisfunction<- function(iris){

Setosa<-iris[which(iris[,"Species"]=="setosa"),]

Virginica<-iris[which(iris[,"Species"]=="virginica"),]

Versicolor<-iris[which(iris[,"Species"]=="versicolor"),]

return(list(Setosa, Virginica, Versicolor))

}

#Then input >Irisfunction(iris)

1. Write a function that takes iris as its argument. The function should, for each row, add **Sepal.Length** and **Petal.Length** if **Sepal.Width** is > 3.1. It should substract **Petal.Length** from **Sepal.Length** if **Sepal.Width** is <3.1. The answer should be returned as a vector.

#Was not able to get this one done (the *if* condition threw me off)

1. Load the mtcars dataset we used in the earlier tests. Write a function that takes a number of cylinders as its argument. Have the function return the average miles per gallon (column **mpg**) for all cars with that many cylinder (column **cyl**).

>CarsFunction<-function(cyl=6){

subset<-mtcars[which(mtcars[,”cyl”]==Cylinder),]

MeanMPG<-mean(Subset[,”mpg”])

Return(MeanMPG)

}

>CarsFunction(cyl)

1. Write a function that simulates 1,000,000 powerball drawings. A powerball drawing takes a random **sample** of 5 numbers (without replacement) from 1 through 69, plus one powerball number ranging from 1 through 26. The function should return a single object recording all of your draws.

>MyPowerball<-matrix(NA, nrow=1000000, ncol=6)

>For (ball in 1:1000000){

MyPowerball[ball,]<- c(Sample(1:69,5,replace=FALSE), Sample(1:26,1,replace=FALSE))

}

>MyPowerball #returns huge matrix

1. Write a function that take a single set of lottery numbers (as a vector) as its **argument**. As before, write a function that simulates 1,000,000 powerball drawings. Have the function return a TRUE or FALSE value if you won any of the drawings.

>MyPowerballVector<- c(NA)

>MyPowerball<- matrix(NA, nrow=1000000, ncol=6)

>for (ball in 1:1000000{

MyPowerball[ball,]<- c(Sample(1:69,5,replace=FALSE), Sample(1:26,1,replace=FALSE))

}

>for (ticket in 1:1000000){

MyPowerballVector[ticket,]<- c(MyPowerballVector) #I know this is wrong

Win<- identical(MyPowerballVector,MyPowerball)

Return(Win)

}

>Win #should return true or false—not sure how to get a function to take the vector

Advanced Questions

Load in the precip dataset.

1. What is the **mean**, **median**, and **standard deviation** of precip?

Mean=34.99, median=36.6, sd=13.21889

1. Is precip best visualized using a barplot( ) or hist( )? Why?

A barplot would be a better representation of the data since each city has only one value. For example, a histogram would be used if we wanted to look at one city and how many years it had a certain amount of rainfall.

1. Generate a vector of random numbers drawn from a normal distribution with the same mean, standard deviation, and number of elements as in the precip dataset. Name this vector RandomNormal.

>RandomNormal<-rnorm(70, mean=34.99, sd=13.21889)

1. Write a function that tests, based on the means of each distribution, whether it is likely that RandomNormal and precip were drawn from the same underlying distribution.

>compareMeans<- function(RandomNormal, precip, Iterations=100){

Both<- c(RandomNormal,precip)

ReplicatedMeans<- array(data=NA, dim=Iterations)

For(Counter in 1:Iterations){

NewRandomNormal<- sample(Both,dim(RandomNormal), replace=FALSE)

Newprecip<- sample(Both, dim(precip), replace=FALSE)

RandomNormalMean<- mean(NewRandomNormal)

PrecipMean<- mean(Newprecip)

ReplicatedMeans[Counter]<- RancomNormalMean-PrecipMean

}

return(ReplicatedMeans)

}

>set.seed(10)

>ReplicationResults<- compareMeans(RandomNormal, precip, 100)

>length(which(ReplicationResults>34.99))

1. Create a density( ) plot of precip and RandomNormal. Is the test you performed above (question 4) a good or bad indicator of whether the two distributions are identical? Why or why not?