2347214
Aravind Nandakumar
MCA - B
Christ University

Task 2

Applied Statistics using R

```
#Create two matrices, matrix_A and matrix_B
matrix_A<-matrix(c(10,5,3,6,3,4,2,5,2),nrow = 3,ncol=3,byrow =
TRUE,dimnames = list(c("mAR1","mAR2","mAR3"),c("mAC1","mAC2","mAC3")))
matrix_B<-matrix(c(2,4,1,9,4,2,7,6,4),nrow = 3,ncol=3,byrow =
TRUE,dimnames = list(c("mBR1","mBR2","mBR3"),c("mBC1","mBC2","mBC3")))

#Calculate the sum of matrix_A and matrix_B and store the result in a new
matrix named matrix_sum.
matrix_sum<-matrix_A+matrix_B
rownames(matrix_sum)=c("m(A+B)R1","m(A+B)R2","m(A+B)R3")
colnames(matrix_sum)=c("m(A+B)C1","m(A+B)C2","m(A+B)C3")
print("Matrix Sum")
print(matrix_sum)

#Calculate the difference between matrix_A and matrix_B and store the
result in a new matrix named matrix_diff.
matrix_diff<-matrix_A-matrix_B
rownames(matrix_diff)=c("m(A-B)R1","m(A-B)R2","m(A-B)R3")
colnames(matrix_diff)=c("m(A-B)C1","m(A-B)C2","m(A-B)C3")
print("Matrix Diffrence")
print(matrix_diff)

#Multiply matrix_A by a scalar value of 2 and store the result in a new
matrix named matrix_mult
matrix_mult<-matrix_A*matrix_B
rownames(matrix_mult)=c("m(A.B)R1","m(A.B)R2","m(A.B)R3")
colnames(matrix_mult)=c("m(A.B)C1","m(A.B)C2","m(A.B)C3")
print("Matrix Multiplication")
print(matrix_mult)
```

```
#Calculate the product of matrix_A and matrix_B and store the result in a
new matrix named matrix_product
matrix_product<-matrix_A %% matrix_B
rownames(matrix_product)=c("m(A*B)R1","m(A*B)R2","m(A*B)R3")
colnames(matrix_product)=c("m(A*B)C1","m(A*B)C2","m(A*B)C3")
print("Matrix Product")
print(matrix_product)


#Find the transpose of matrix_A and store the result in a new matrix named
matrix_A_transpose
matrix_A_transpose<-t(matrix_A)
print("Matrix A Transpose")
print(matrix_A_transpose)


#Calculate the determinant of matrix_B and store it in a variable named
determinant_B
determinant_B<-det(matrix_B)
print("Matrix B Determinant")
print(determinant_B)


#Invert matrix_B to obtain the inverse matrix and store it in a new matrix
named matrix_B_inverse
matrix_B_inverse<-solve(matrix_B)
print("Matrix B Inverse")
print(matrix_B_inverse)


#Check if matrix_B is orthogonal (i.e., its transpose is equal to its
inverse)
print("Checking If Matrix B is Orthogonal ...")
matrix_B_transpose<-t(matrix_B)
orthogonal_B<-matrix_B_transpose==matrix_B_inverse
c=0
for (x in orthogonal_B) {
  if(x==FALSE){
    c=1
    break
```

```
  }
}
if(c==0){
  print("It is orthogonal")
}else{
  print("It is not orthogonal")
}


#Calculate the element-wise square root of matrix_A and store the result
in a new matrix named matrix_A_sqrt
matrix_A_sqrt<-sqrt(matrix_A)
print("SquareRoot of Matrix A")
print(matrix_A_sqrt)


#Calculate the mean of all the elements in matrix_B
mean_matrix_B<-mean(matrix_B)
print("Mean of Matrix A ")
print(mean_matrix_B)


#Calculate the sum of each column in matrix_A.
sum_matrix_A<-c(sum(matrix_A[,1]),sum(matrix_A[,2]),sum(matrix_A[,3]))
print("Matrix A Colomn Sum")
print(sum_matrix_A)

#Calculate the row means of matrix_B.
row_means_matrix_B<-c(mean(matrix_B[1,]),mean(matrix_B[2,]))
print("Matrix B Row Mean")
print(row_means_matrix_B)

#Extract the second row of matrix_A and store it in a vector named
second_row_A
second_row_A<-matrix_A[2,]
print("Second Row of Matrix A")
print(second_row_A)

#Extract the third column of matrix_B and store it in a vector named
third_column_B
```

```
third_column_B<-matrix_B[,3]
print("Third Column of Matrix B")
print(third_column_B)
```

Output

```
> #Create two matrices, matrix_A and matrix_B
> matrix_A<-matrix(c(10,5,3,6,3,4,2,5,2),nrow = 3,ncol=3,byrow = TRUE,dimnames
= list(c("mAR1","mAR2","mAR3"),c("mAC1","mAC2","mAC3")))
> matrix_B<-matrix(c(2,4,1,9,4,2,7,6,4),nrow = 3,ncol=3,byrow = TRUE,dimnames
= list(c("mBR1","mBR2","mBR3"),c("mBC1","mBC2","mBC3")))
>
> #Calculate the sum of matrix_A and matrix_B and store the result in a new
matrix named matrix_sum.
> matrix_sum<-matrix_A+matrix_B
> rownames(matrix_sum)=c("m(A+B)R1","m(A+B)R2","m(A+B)R3")
> colnames(matrix_sum)=c("m(A+B)C1","m(A+B)C2","m(A+B)C3")
> print("Matrix Sum")
[1] "Matrix Sum"
> print(matrix_sum)
         m(A+B)C1 m(A+B)C2 m(A+B)C3
m(A+B)R1       12        9        4
m(A+B)R2       15        7        6
m(A+B)R3        9       11        6
>
> #Calculate the difference between matrix_A and matrix_B and store the result
in a new matrix named matrix_diff.
> matrix_diff<-matrix_A-matrix_B
> rownames(matrix_diff)=c("m(A-B)R1","m(A-B)R2","m(A-B)R3")
> colnames(matrix_diff)=c("m(A-B)C1","m(A-B)C2","m(A-B)C3")
> print("Matrix Diffrence")
[1] "Matrix Diffrence"
> print(matrix_diff)
         m(A-B)C1 m(A-B)C2 m(A-B)C3
m(A-B)R1        8        1        2
m(A-B)R2       -3       -1        2
m(A-B)R3       -5       -1       -2
>
> #Multiply matrix_A by a scalar value of 2 and store the result in a new
matrix named matrix_mult
> matrix_mult<-matrix_A*matrix_B
> rownames(matrix_mult)=c("m(A.B)R1","m(A.B)R2","m(A.B)R3")
> colnames(matrix_mult)=c("m(A.B)C1","m(A.B)C2","m(A.B)C3")
> print("Matrix Multiplication")
```

```
[1] "Matrix Multiplication"
> print(matrix_mult)
         m(A.B)C1 m(A.B)C2 m(A.B)C3
m(A.B)R1       20       20        3
m(A.B)R2       54       12        8
m(A.B)R3       14       30        8
>
> #Calculate the product of matrix_A and matrix_B and store the result in a
new matrix named matrix_product
> matrix_product<-matrix_A %% matrix_B
> rownames(matrix_product)=c("m(A*B)R1","m(A*B)R2","m(A*B)R3")
> colnames(matrix_product)=c("m(A*B)C1","m(A*B)C2","m(A*B)C3")
> print("Matrix Product")
[1] "Matrix Product"
> print(matrix_product)
         m(A*B)C1 m(A*B)C2 m(A*B)C3
m(A*B)R1        0        1        0
m(A*B)R2        6        3        0
m(A*B)R3        2        5        2
>
>
> #Find the transpose of matrix_A and store the result in a new matrix named
matrix_A_transpose
> matrix_A_transpose<-t(matrix_A)
> print("Matrix A Transpose")
[1] "Matrix A Transpose"
> print(matrix_A_transpose)
     mAR1 mAR2 mAR3
mAC1   10    6    2
mAC2    5    3    5
mAC3    3    4    2
>
>
> #Calculate the determinant of matrix_B and store it in a variable named
determinant_B
> determinant_B<-det(matrix_B)
> print("Matrix B Determinant")
[1] "Matrix B Determinant"
> print(determinant_B)
[1] -54
>
>
> #Invert matrix_B to obtain the inverse matrix and store it in a new matrix
named matrix_B_inverse
> matrix_B_inverse<-solve(matrix_B)
> print("Matrix B Inverse")
[1] "Matrix B Inverse"
> print(matrix_B_inverse)
            mBR1        mBR2        mBR3
```

```
mBC1 -0.07407407  0.18518519 -0.07407407
mBC2  0.40740741 -0.01851852 -0.09259259
mBC3 -0.48148148 -0.29629630  0.51851852
>
>
> #Check if matrix_B is orthogonal (i.e., its transpose is equal to its
inverse)
> print("Checking If Matrix B is Orthogonal ...")
[1] "Checking If Matrix B is Orthogonal ..."
> matrix_B_transpose<-t(matrix_B)
> orthogonal_B<-matrix_B_transpose==matrix_B_inverse
> c=0
> for (x in orthogonal_B) {
+   if(x==FALSE){
+     c=1
+     break
+   }
+ }
> if(c==0){
+   print("It is orthogonal")
+ }else{
+   print("It is not orthogonal")
+ }
[1] "It is not orthogonal"
>
>
> #Calculate the element-wise square root of matrix_A and store the result in
a new matrix named matrix_A_sqrt
> matrix_A_sqrt<-sqrt(matrix_A)
> print("SquareRoot of Matrix A")
[1] "SquareRoot of Matrix A"
> print(matrix_A_sqrt)
         mAC1     mAC2     mAC3
mAR1 3.162278 2.236068 1.732051
mAR2 2.449490 1.732051 2.000000
mAR3 1.414214 2.236068 1.414214
>
>
> #Calculate the mean of all the elements in matrix_B
> mean_matrix_B<-mean(matrix_B)
> print("Mean of Matrix A ")
[1] "Mean of Matrix A "
> print(mean_matrix_B)
[1] 4.333333
>
>
> #Calculate the sum of each column in matrix_A.
> sum_matrix_A<-c(sum(matrix_A[,1]),sum(matrix_A[,2]),sum(matrix_A[,3]))
> print("Matrix A Colomn Sum")
```

```
[1] "Matrix A Colomn Sum"
> print(sum_matrix_A)
[1] 18 13  9
>
> #Calculate the row means of matrix_B.
> row_means_matrix_B<-c(mean(matrix_B[1,]),mean(matrix_B[2,]))
> print("Matrix B Row Mean")
[1] "Matrix B Row Mean"
> print(row_means_matrix_B)
[1] 2.333333 5.000000
>
> #Extract the second row of matrix_A and store it in a vector named
second_row_A
> second_row_A<-matrix_A[2,]
> print("Second Row of Matrix A")
[1] "Second Row of Matrix A"
> print(second_row_A)
mAC1 mAC2 mAC3
   6    3    4
>
> #Extract the third column of matrix_B and store it in a vector named
third_column_B
> third_column_B<-matrix_B[,3]
> print("Third Column of Matrix B")
[1] "Third Column of Matrix B"
> print(third_column_B)
mBR1 mBR2 mBR3
   1    2    4
```