实 验 报 告

学 院：

课程名称：

专 业：

姓 名：

学 号：

1. 实验目的

掌握数据新变量的生成；

掌握层次分析法的代码实现；掌握个体相似度计算；

掌握函数的编写。

1. 实验内容
   1. 使用如下矩阵作为判断矩阵，根据“地区总人口”、“性别比”、“15-59岁人口比重”、“15岁以上人口受教育年限”这四个变量matrix(c(1,5,3,7,1/5,1,1/3,5,1/3,3,1,7,1/7,1/5,1/7,1),nrow=4,byrow=T)，采用层次分析法对各省份进行评分。代码中需要包括一致性检验，计算特征向量和权重，计算各省份最终得分。
   2. 编写函数名为sim\_value的函数，使用夹角余弦来计算某省与其他省份之间的相似度，评判变量为“地区总人口”、“性别比”、“15-59岁人口比重”、“15岁以上人口受教育年限”。请在函数中设置参数prov来表示某省，即调用函数时可以任意指定某省份，计算其与其他省份的“距离”。
2. 实验要求

完成实验内容中的每一步，请附上代码。2.1中对最后结果进行排序，然后汇报结果的前10行；2.2中调用函数计算河南省与其他省份的距离，排序后，汇报前10行。

# 代码

library("readxl")

library("dplyr")

library("magrittr")

#4.2.1

#准则层

#C1=“地区总人口

#C2=“性别比”

#C3=“15-59岁人口比重”

#c4=15岁以上人口受教育年限”

#方案层

#Fi

t<-total\_info[-28,]

row.names(t)<-c(1:31)

t\_c1<-as.double(t$地区总人口)

t\_c2<-t$性别比

t\_c3<-t$X15.59岁

t\_c4<-t$X15岁以上人口受教育年限

a2m<-function(t){

m<-matrix(0,nrow = 31,ncol = 31)

for (i in c(1:31)){

for(j in c(1:31)){

m[i,j]=t[i]/t[j]

}

}

return(m)

}

data\_F1<-a2m(t\_c1)

data\_F2<-a2m(t\_c2)

data\_F3<-a2m(t\_c3)

data\_F4<-a2m(t\_c4)

#准则层判断矩阵

data\_C <- matrix(c(1,5,3,7,1/5,1,1/3,5,1/3,3,1,7,1/7,1/5,1/7,1),nrow =4,dimnames = list(c("C1","C2","C3","C4"),c("C1","C2","C3","C4"))

)

#归一

Weigth\_fun <- function(data){

if(class(data) == 'matrix'){

data = data

} else {

if ( class(data) == 'data.frame' & nrow(data) == ncol(data) - 1 & is.character(data[,1,drop = TRUE])){

data = as.matrix(data[,-1])

} else if (class(data) == 'data.frame' & nrow(data) == ncol(data)) {

data = as.matrix(data)

} else {

stop('please recheck your data structure , you must keep a equal num of the row and col')

}

}

sum\_vector\_row = data %>% apply(2,sum)

decide\_matrix = data %>% apply(1,function(x) x/sum\_vector\_row)

weigth\_vector = decide\_matrix %>% apply(2,sum)

result = list(decide\_matrix = decide\_matrix, weigth\_vector = weigth\_vector/sum(weigth\_vector ))

return(result)

}

Weigth\_fun(data\_C)

#特征向量λ

AW\_Weight <- function(data){

if(class(data) == 'matrix'){

data = data

} else {

if ( class(data) == 'data.frame' & nrow(data) == ncol(data) - 1 & is.character(data[,1,drop = TRUE])){

data = as.matrix(data[,-1])

} else if (class(data) == 'data.frame' & nrow(data) == ncol(data)) {

data = as.matrix(data)

} else {

stop('please recheck your data structure , you must keep a equal num of the row and col')

}

}

AW\_Vector = data %\*% Weigth\_fun(data)$weigth\_vector

λ = (AW\_Vector/Weigth\_fun(data)$weigth\_vector) %>% sum(.) %>% `/`(length(AW\_Vector))

result = list(

AW\_Vector = AW\_Vector,

`∑AW/W` = AW\_Vector/Weigth\_fun(data)$weigth\_vector,

λ = λ

)

return(result)

}

AW\_Weight(data\_C)

#一致性

Consist\_Test <- function(λ,n){

RI\_refer = c(0,0,0.52,0.89,1.12,1.26,1.36,1.41,1.46,1.49,1.52,1.54)

CI = (λ - n)/(n - 1)

CR = CI/(RI\_refer[n])

if (CR <= .1){

cat(" 通过一致性检验！",sep = "\n")

cat(" Wi: ", round(CR,4), "\n")

} else {

cat(" 请调整判断矩阵！","\n")

}

return(CR)

}

Consist\_Test(AW\_Weight(data\_C)$λ,4)

#排序

rule\_Weigth\_C <- Weigth\_fun(data\_C)$weigth\_vector #准则层特征向量

rule\_Weigth\_f1 <- Weigth\_fun(data\_F1)$weigth\_vector #方案层(for f1)特征向量

rule\_Weigth\_f2 <- Weigth\_fun(data\_F2)$weigth\_vector #方案层(for f2)特征向量

rule\_Weigth\_f3 <- Weigth\_fun(data\_F3)$weigth\_vector #方案层(for f3)特征向量

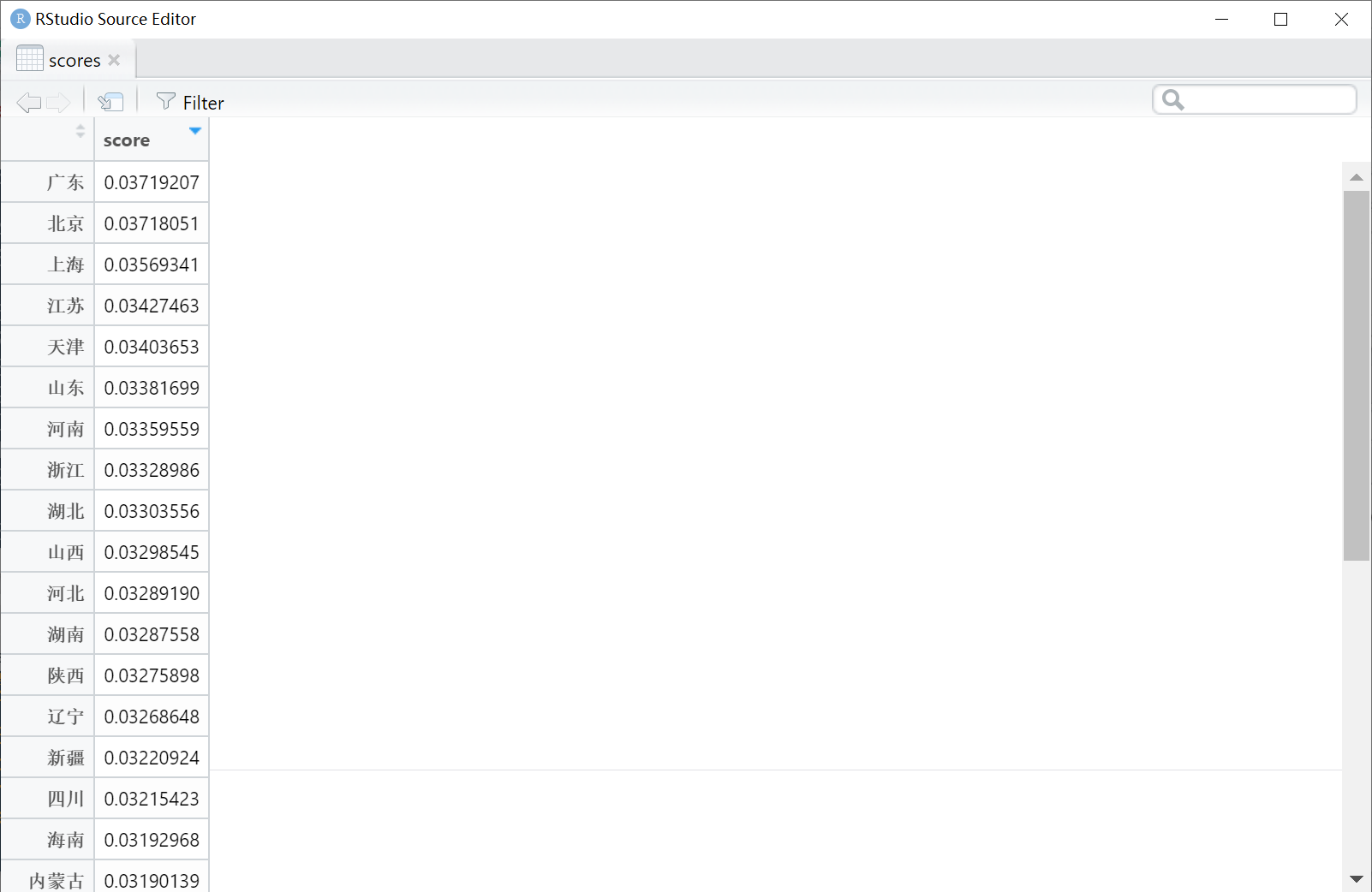
rule\_Weigth\_f4 <- Weigth\_fun(data\_F4)$weigth\_vector #方案层(for f4)特征向量

层次总排序：

all\_matrix <- matrix(c(rule\_Weigth\_f1,rule\_Weigth\_f2,rule\_Weigth\_f3,rule\_Weigth\_f4),nrow = 31)

scores <- all\_matrix %\*% rule\_Weigth\_C

dimnames(scores) <- list(t$province,"score")

view(scores) 

#4.2.2

sim\_value<-function(prov){

vct<-matrix(data=c(t\_c1,t\_c2,t\_c3,t\_c4),ncol = 4,dimnames = list(t$province,c("pop","sexrt","yonung","edu")))

A=as.vector(vct[prov,])

prov\_n<-which(row.names(vct)==prov)

options(digits=17)

cosθ<-data.frame(t$province)

for (i in c(1:31)){

if (i!=prov\_n){

B=as.vector(vct[i,])

cosθ[i,2] <- as.double(sum(A\*B)/(sqrt(sum(A^2))\*sqrt(sum(B^2))))

}

else{next}

}

as.double(cosθ[,2])

colnames(cosθ)[2]<-'Distance'

cosθ<- cosθ[order(cosθ[,2],decreasing = T),]

return(cosθ)

}

options(digits=17)

H<-sim\_value('河南')

view(H)

