qbs124_hw4_gibran

Gibran Erlangga

4/23/2022

Question 1

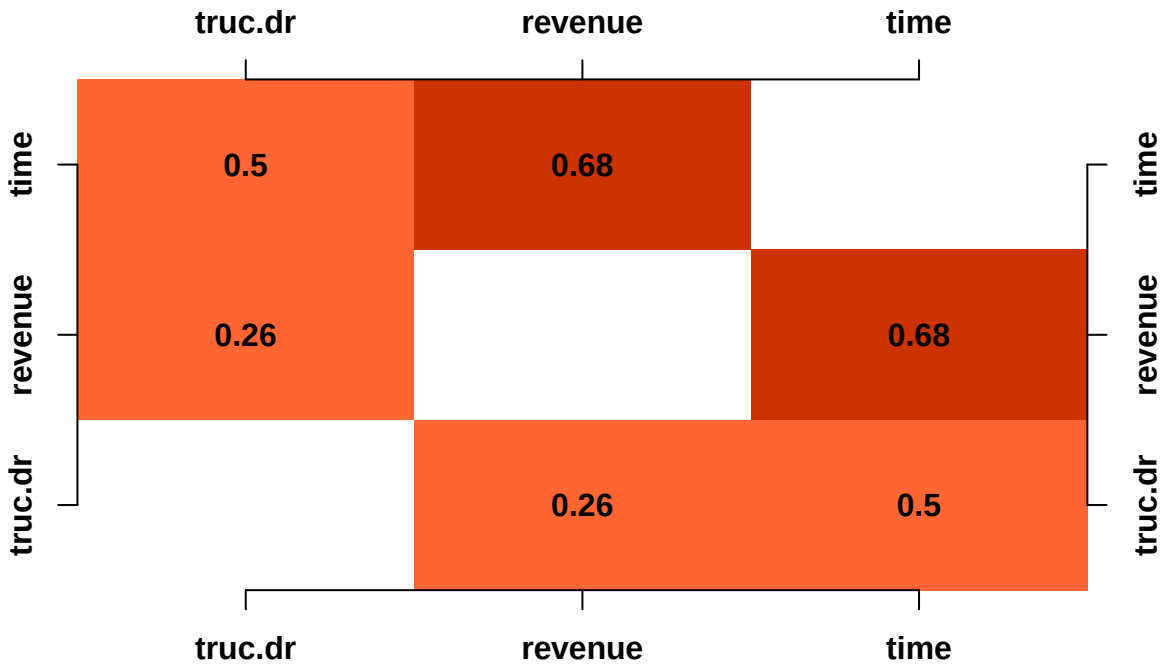
(10 points). Compute the 3 × 3 partial correlation matrix for Revenue and truck drivers example using two methods: by inverse correlation matrix and correlation of residuals. Make sure that the two matrices coincide. Interpret the result in layman terms.

```
# setup
data <- read.csv('truckR.data.csv')
head(data, 2)
```

```
##     truc.dr  revenue
## 1 0.4380666 199.5622
## 2 1.5074743 201.2457
```

```
# add time variable
data$time = 1:nrow(data)

cor_matrix <- cor(data)
iR=iiR=solve(cor_matrix)
ns = length(data)
column_name = names(data)
```

```
# inverse correlation
# based on cimcorSP(job=3)
for(i in 1:ns)
  for(j in 1:ns)
    iR[i,j]=-iiR[i,j]/sqrt(abs(iiR[i,i]*iiR[j,j]))
    diag(iR)=rep(NA,ns)
    image(1:ns,1:ns,iR,col=c("cyan","#ff6633","#cc3300"),
          breaks=c(0,.25,.5,.75), xlab="",ylab="",axes=F)
    axis(side=1,at=1:ns,labels=column_name,font=2)
    axis(side=2,at=1:ns,labels=column_name,srt=90,font=2)
    axis(side=3,at=1:ns,labels=column_name,font=2)
    axis(side=4,at=1:ns,labels=column_name,font=2)
    for(i in 1:ns)
        text(rep(i,ns),1:ns,round(iR[i,],2),font=2)
```




```
# correlation of residuals

# reference: https://towardsdatascience.com/keeping-an-eye-on-confounds-a-walk-through-for-calculating-

partialCors <- list()
for (i in 1:length(column_name)) {
  y <- column_name[i]
  covariatesAll <- column_name[!(column_name %in% y)]
  crntPcor <- double()
  for (j in 1:length(covariatesAll)) {
    covarLeftOut <- covariatesAll[j]
    covariatesCrnt <- covariatesAll[!(covariatesAll %in% covarLeftOut)]

    # construct lm model
    rhs <- paste(covariatesCrnt, collapse = " + ")
    lhs <- paste(y, "~")
    frmla <- as.formula(paste(lhs, rhs))

    # get residuals from linear model of X given Z
    R1 <- lm(frmla, data = data)$residual
    lhs <- paste(covarLeftOut, "~")
    frmla <- as.formula(paste(lhs, rhs))

    # get residuals from linear model of Y given Z
    R2 <- lm(frmla, data = data)$residual
    crntPcor[j] <- cor(R1,R2)
```

```
  }
  partialCors[[i]] <- append(crntPcor, 1 , i-1)
}

partialCorMat <- matrix(unlist(partialCors),
                        ncol = length(column_name),
                        nrow = length(column_name),
                        dimnames = list(column_name, column_name))
iR<-partialCorMat
ns=3
par(mfrow=c(1,1),mar=c(0,1,2,1))

diag(iR)=rep(NA,ns)
    image(1:ns,1:ns,iR,col=c("yellow","goldenrod1","pink2"),
          breaks=c(0,.25,.5,.75),xlab="",ylab="",axes=F)
    axis(side=1,at=1:ns,labels=column_name,font=2)
    axis(side=2,at=1:ns,labels=column_name,srt=90,font=2)
    axis(side=3,at=1:ns,labels=column_name,font=2)
    axis(side=4,at=1:ns,labels=column_name,font=2)
    for(i in 1:ns)
        text(rep(i,ns),1:ns,round(iR[i,],2),font=2)
```

Interpretation:

Both plots above show the exact same result with different color coding. We can see from both plots that we can get partial correlation matrix by computing inverse correlation matrix and correlation of residuals.

Question 2

(5 points). Explain in layman language false correlation referring to Example 1.

Answer: False correlation in above example happened in the correlation value between Truck Drivers and Revenue, which shows a high correlation value. The number of truck drivers should not necessarily increase the revenue of the company. This happened because we considered both variables without accounting for time variable. To solve it, we use partial correlation. In partial correlation, we can see that there is low correlation between both variables. In conclusion, the original correlation plot is misnomer and hence we should apply partial correlation to get the true relationship between these variables.

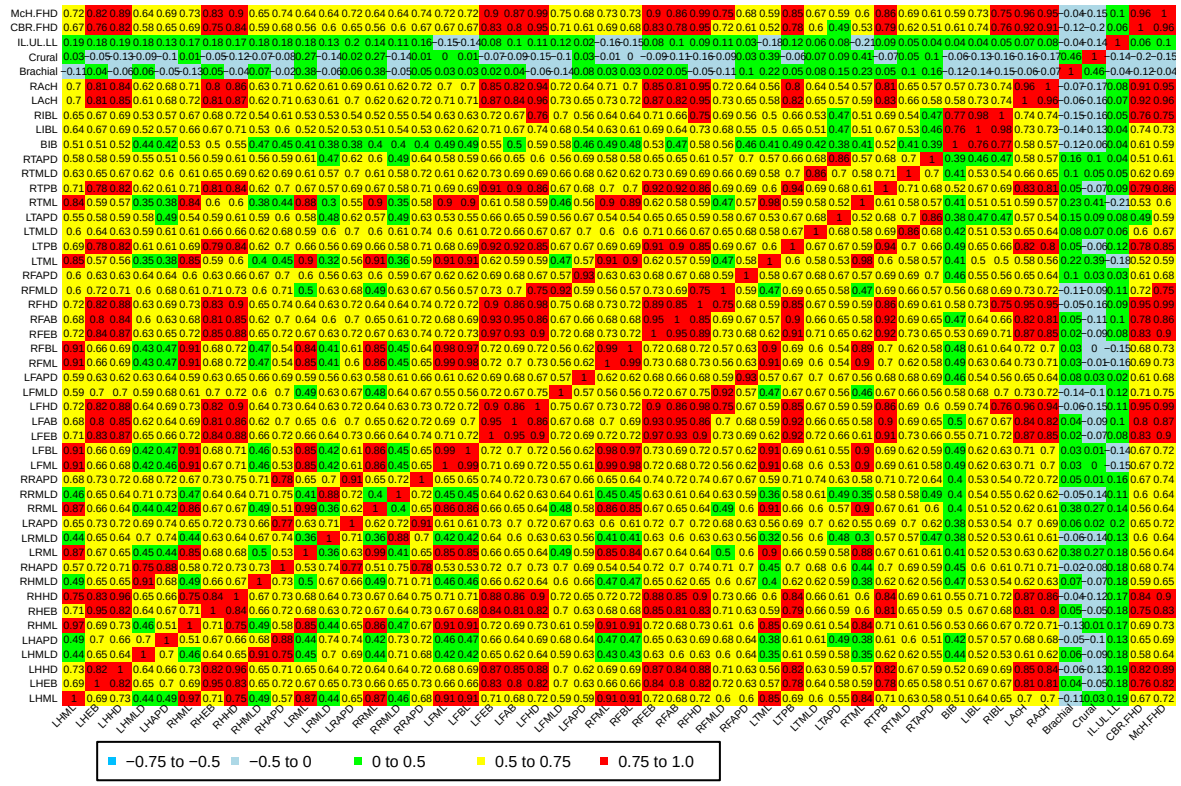
Question 3

(10 points). (a) Remove the columns that have -1 or 1 correlation with others as follows from skelet_2.pdf.

```
d <- read.csv('Goldman.csv')

# remove cor = 1
d <- subset(d, select = -c(AVG.FHD, GRINE.FHD))
# remove cor = -1
d <- subset(d, select = -c(IL.LL.UL))

# setup
female=as.numeric(d[,3])
```

```
## Warning: NAs introduced by coercion
```

```
d=d[,18:ncol(d)]
nm=names(d)
nc=ncol(d);nr=nrow(d)
for(i in 1:nc)
if(sum(is.na(d[,i]))==nr) {alln=i;break}
d=d[,-alln]
nm=nm[-alln]
nc=ncol(d)
R=cor(d,use="pairwise.complete.obs")
n=nrow(R)

par(mfrow=c(1,1),mar=c(1,1,1,1))
cl=c("deepskyblue","lightblue","green","yellow","red")
image(1:n,1:n,breaks=c(-.75,-.5,0,.5,.75,1),ylim=c(-5,n+1),xlim=c(-2,n+.5),
      col=cl,ylab="",xlab="",R,axes=F)
text(1:n,rep(0.5,n),nm,adj=1,cex=.3,srt=45)
text(rep(.3,n),1:n,nm,adj=1,cex=.3)
for(i in 1:n)
  for(j in 1:n)
  text(i,j,round(R[i,j],2),cex=.3)
  mtext(side=3,paste("Correlation heatmap of",n,"osteometric measurements taken from 1538 human skeleto
  br=c("-0.75 to -0.5","-0.5 to 0","0 to 0.5","0.5 to 0.75","0.75 to 1.0")
  legend(2,-2,br,col=cl,pch=15,horiz=T,cex=.5)
```



(b) Apply option use="complete.obs" when computing the regular correlation matrix and then regularize it by adding 10^{-20} to the diagonal elements.

```
d <- read.csv('Goldman.csv')

# remove cor = 1
d <- subset(d, select = -c(AVG.FHD, GRINE.FHD))
# remove cor = -1
d <- subset(d, select = -c(IL.LL.UL))

# setup
female=as.numeric(d[,3])
```

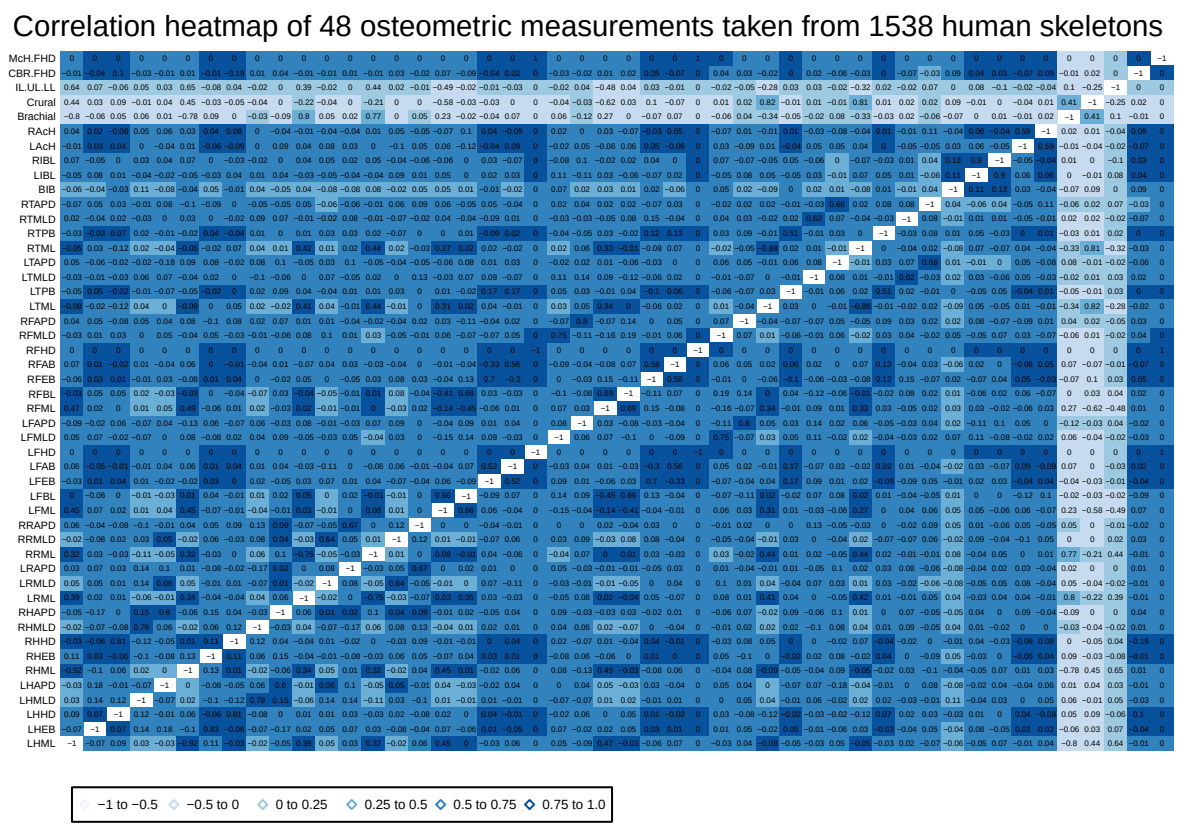
```
## Warning: NAs introduced by coercion
```

```
d=d[,18:ncol(d)]
nm=names(d)
nc=ncol(d);nr=nrow(d)
for(i in 1:nc)
if(sum(is.na(d[,i]))==nr) {alln=i;break}
d=d[,-alln]
nm=nm[-alln]
nc=ncol(d)
R=cor(d,use="complete.obs")
n=nrow(R)
```

```
# add regularization
diag(R) <- diag(R) + 10**-10
```

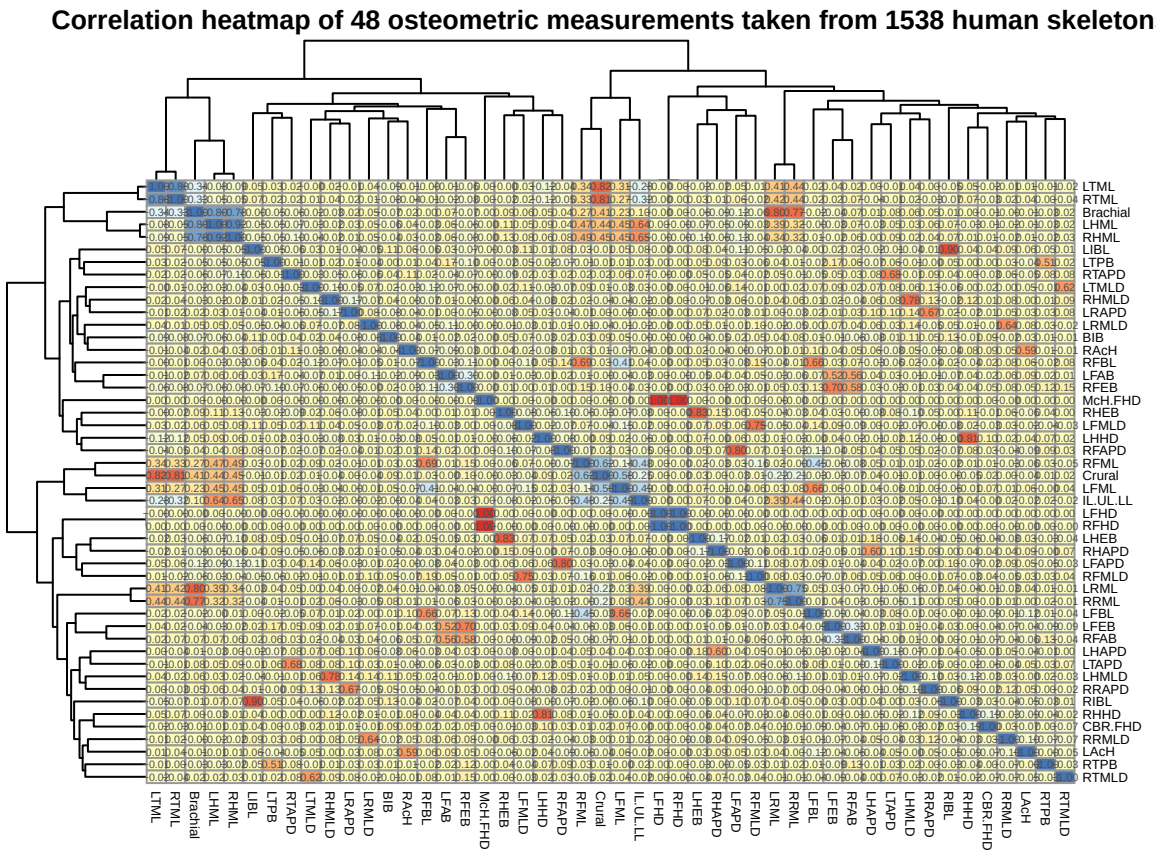
(c) Compute and display the partial correlation matrix. Use your own breaks and colors (see Rcolor.pdf) to cover the range of correlation coefficients from -1 to 1.

```
# (a) but change your threshold and color coding
iR=iiR=solve(R)
ns = length(d)

# Calculating partial correlation matrix
for(i in 1:ns)
  for(j in 1:ns)
    iR[i,j]=-iiR[i,j]/sqrt(abs(iiR[i,i]*iiR[j,j]))

# Assigning colors
par(mfrow=c(1,1),mar=c(0,1,2,1))
cl= RColorBrewer::brewer.pal(6, 'Blues')

image(1:n,1:n,breaks=c(-1, -.5, 0, 0.25,.5,.75,1),ylim=c(-5,n+1),xlim=c(-2,n+.5),col=cl,ylab="",xlab="
text(rep(.3,n),1:n,nm,adj=1,cex=.35)
for(i in 1:n)
  for(j in 1:n)
    text(i,j,round(iR[i,j],2),cex=.25)
    mtext(side=3,paste("Correlation heatmap of",n,"osteometric measurements taken from 1538 human skele
    br=c("-1 to -0.5","-0.5 to 0","0 to 0.25","0.25 to 0.5", "0.5 to 0.75","0.75 to 1.0")
    legend(1,-2,br,col=cl,pch=5,horiz=T,cex=.45)
```



(d) Display the partial correlation matrix using pheatmap package.

```
library(pheatmap)

pheatmap(iR, display_numbers = TRUE,
         fontsize = 8,
         fontsize_number = 4,
         fontsize_col = 5,
         fontsize_row = 5,
         main = "Correlation heatmap of 48 osteometric measurements taken from 1538 human skeletons")
```



(d) Make your interpretation and conclusion. Save the last heatmap in large size png format file.

Interpretation: We see a different result on the original correlation heatmap with the partial correlation heatmap. This happened because tall people tend to have longer measurement of their bones compared to short people, and there appears to be correlation between these variables, as shown in the original correlation heatmap. When we apply partial correlation, the picture is significantly different. This explains that the measurement of bones are not correlated with each other, but it seems to be when they are from the same skeleton.

```
#png(file = 'pheatmap.png', width = 24, height = 24, units = "in", res = 300)
#pheatmap(iR, display_numbers = TRUE,
#         fontsize = 25,
#         fontsize_number = 10,
#         fontsize_col = 12,
#         fontsize_row = 12,
#         main = "Correlation heatmap of 48 osteometric measurements taken from 1538 human skeletons")
#dev.off()
```