Homework Assignment 1

Kevin Chen

6/17/2020

Gene Expression Analysis and Visualization 410.671 HW #1

For this homework, we will be working with a study from Gene Expression Omnibus (GEO) with the accession GDS2880. This is an Affymetrix microarray experiment (HGU133A array). The data researchers were investigating patient matched normal and stage 1 or stage 2 clear cell renal cell carcinoma (cRCC) tumors to provide insight into the molecular pathogenesis of cRCC. We will be conducting outlier analysis using various methods to identify aberrant samples, followed by missing value imputation to assess the accuracy of two different algorithms.

1.) Download and load the renal cell carcinoma data file into R. Make sure that the row names are in the correct location (Affymetrix fragment names). Look at the dimensions and verify that you have 22 arrays and 22,283 probesets. (2pts.)

```
renal<-read.table("renal_cell_carcinoma/renal_cell_carcinoma.txt", header=T, row.names=1)
renal.annotation<-read.table("renal_carcinoma_annotation/renal_carcinoma_annotation.txt", row.names=1)
dim(renal)
```

```
## [1] 22283    22
```

2.) Label the header columns of your data frame maintaining the GSM ID, but adding the Normal/Tumor identity. (2pts.)

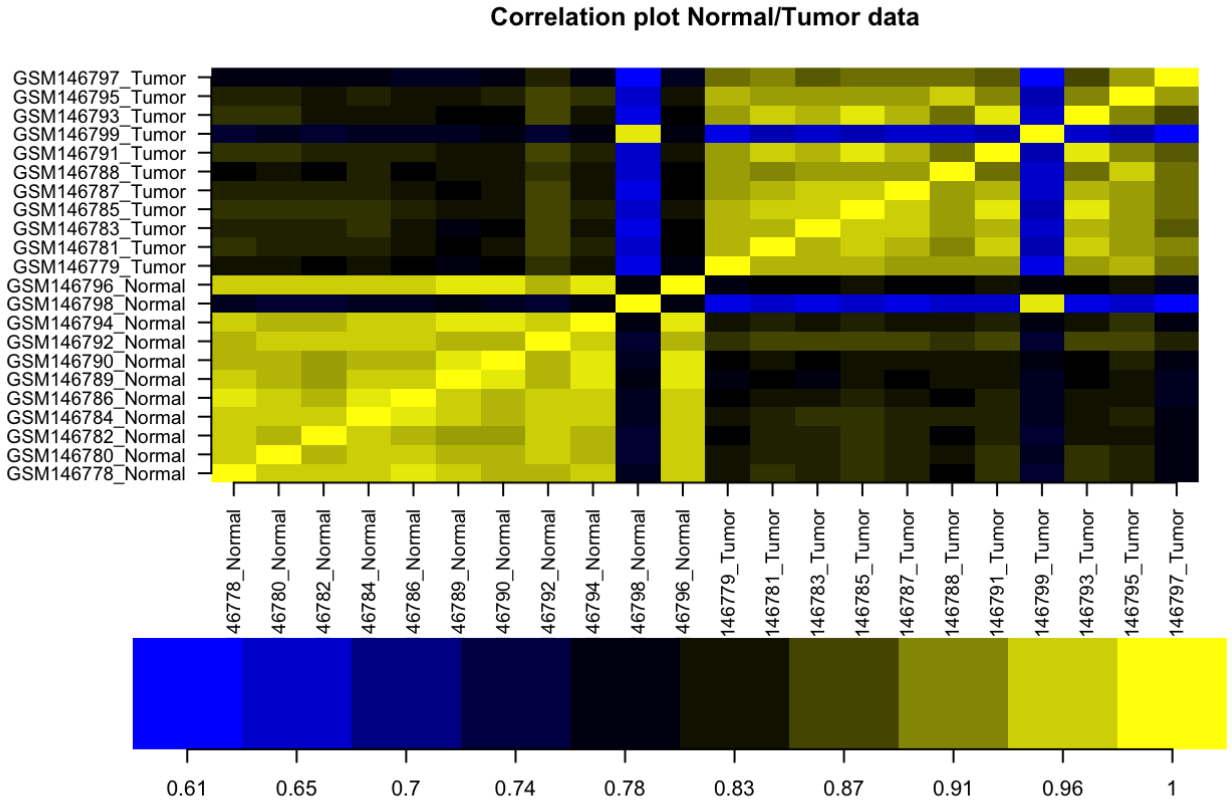
```
colnames(renal)<-sapply(colnames(renal),function(z) {paste0(z,"_",renal.annotation[z,"V9"])})
```

3.) Identify any outlier samples using the following visual plots: 4.) Correlation plot (heat map) (2pts.)

```
library(gplots)
dat.cor <- cor(renal)
layout(matrix(c(1,1,1,1,1,1,1,1,2,2), 5, 2, byrow = TRUE))
par(oma=c(5,7,1,1))
cx <- rev(colorpanel(25,"yellow","black","blue"))
leg <- seq(min(dat.cor,na.rm=T),max(dat.cor,na.rm=T),length=10)
image(dat.cor,main="Correlation plot Normal/Tumor data",axes=F,col=cx)
```

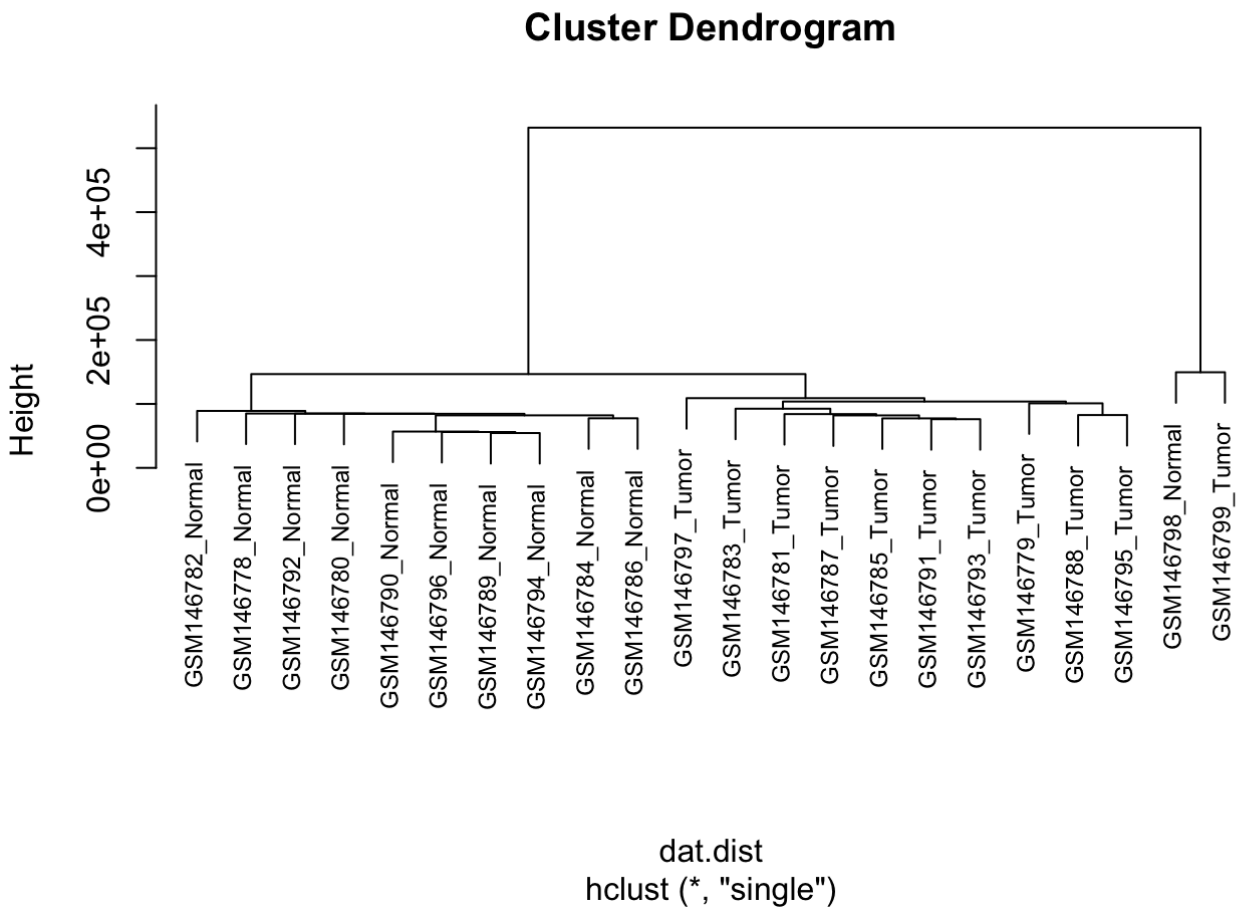
```
axis(1,at=seq(0,1,length=ncol(dat.cor)),label=dimnames(dat.cor)[[2]],cex.axis=0.9,las=2)

axis(2,at=seq(0,1,length=ncol(dat.cor)),label=dimnames(dat.cor)[[2]],cex.axis=0.9,las=2)
par(mar=c(1,1,1,1))
image(as.matrix(leg),col=cx,axes=F)
tmp <- round(leg,2)
axis(1,at=seq(0,1,length=length(leg)),labels=tmp,cex.axis=1)
```



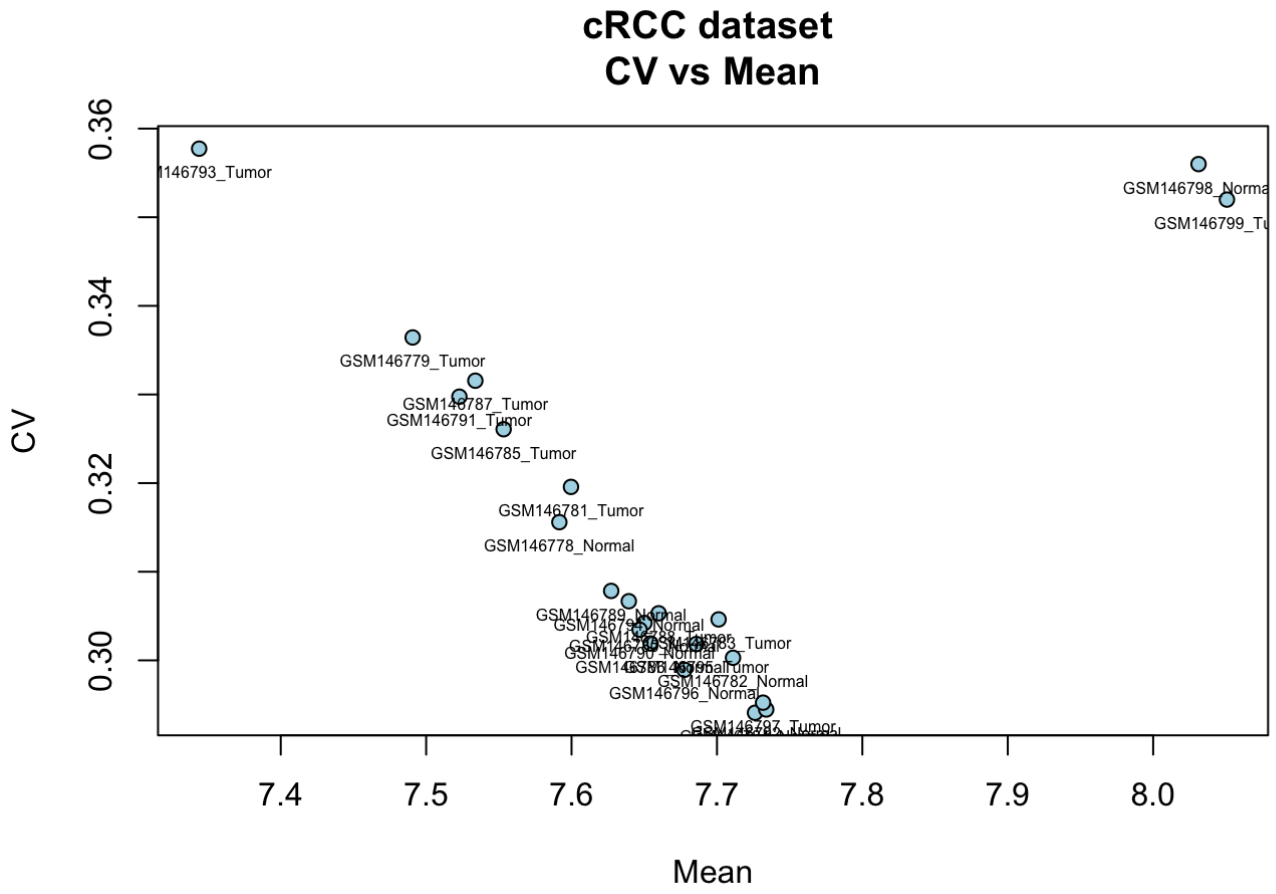
Hierarchical clustering dendrogram (2pts.)

```
dat<-renal
dat <- t(dat) #transpose dat
dat.dist <- dist(dat,method="euclidean")
dat.clust <- hclust(dat.dist,method="single")
plot(dat.clust,labels=names(dat),cex=0.75)
```



CV vs. mean plot (2pts.)

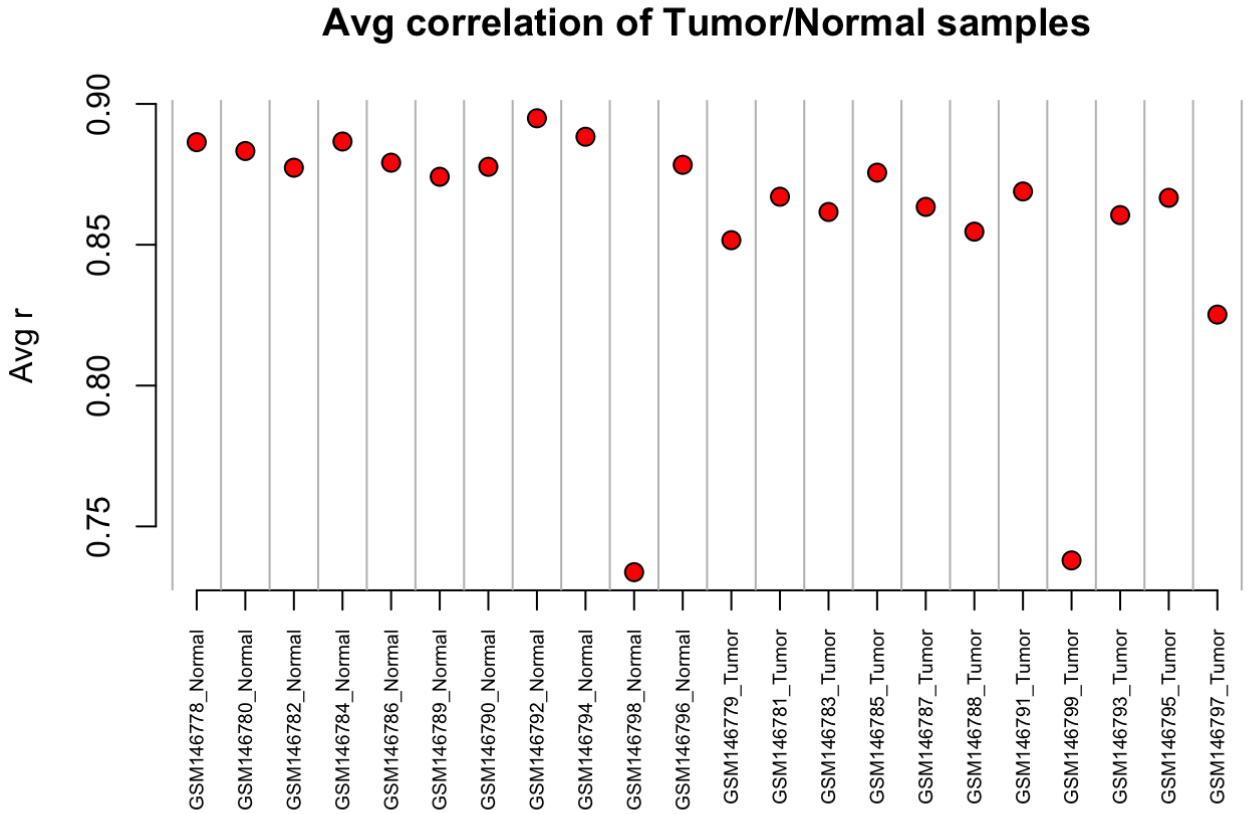
```
dat<-renal
dat.mean <- apply(log2(dat),2,mean)
dat.sd <- sqrt(apply(log2(dat),2,var))
dat.cv <- dat.sd/dat.mean
plot(dat.mean,dat.cv,main="cRCC dataset \nCV vs Mean",xlab="Mean",ylab="CV",col='blue',cex=1.5,type="n")
points(dat.mean,dat.cv,bg="lightblue",col=1,pch=21)
text(dat.mean,dat.cv,label=dimnames(dat)[[2]],pos=1,cex=0.5)
```



Average correlation plot (2pts.)

```
dat.avg <- apply(dat.cor,1,mean)
par(oma=c(3,0.1,0.1,0.1))
plot(c(1,length(dat.avg)),range(dat.avg),type="n",xlab="",ylab="Avg r",main="Avg correlation of Tumor/Normal samp
les",axes=F)
points(dat.avg,bg="red",col=1,pch=21,cex=1.25)

axis(1,at=c(1:length(dat.avg)),labels=dimnames(dat)[[2]],las=2,cex.lab=0.4,cex.axis=0.6)
axis(2)
abline(v=seq(0.5,62.5,1),col="grey")
```



For all plots, make sure you label the points appropriately, title plots, and label axes. You will also need to provide a legend for the correlation plot. You can use the gplots for a color gradient, or just use the default colors. 6.) Install and load the impute library. (1pt.)

```
BiocManager::install("impute")
library(impute)
```

7.) Remove the outlier samples you identified in the first part of this assignment. (2pts.)

```
# based on the previous 4 plots, we consistently see that GSM146798 (Normal) and GSM146799 (Tumour) show abberant
 values compared to the other data points across all of the plots. This abnormality is unlikely to be due to unde
rlying biological processes because GSM146798, a normal sample, show low correlation with othe normal samples in
 the heat map. Likewise, GSM146799, a tumour sample, also does not show strong correlation with other tumour samp
les in the heat map. Thus, it is fair to assume that they are outliers. We should therefore remove them.

remove<-which(colnames(renal) %in% c("GSM146798_Normal","GSM146799_Tumor"))
renal<-renal[,-remove]
```

8.) Now we are going to use a couple of transcripts that were determined in this study to be indicative of normal renal function. The genes we will assess are kininogen 1 (KNG1) and aquaporin 2 (AQP2). Using either NetAffx or Gene Cards websites (or other resources, if you like), extract the probesets for these two genes. Hint: KNG1 has two while AQP2 has one. Then plot a profile plot (expression intensity vs. samples) for each probeset for these two genes. You may have to convert the data frame row to a vector to plot it. Do the plots of these genes seem to indicate normal renal function? Explain. (6pts.)

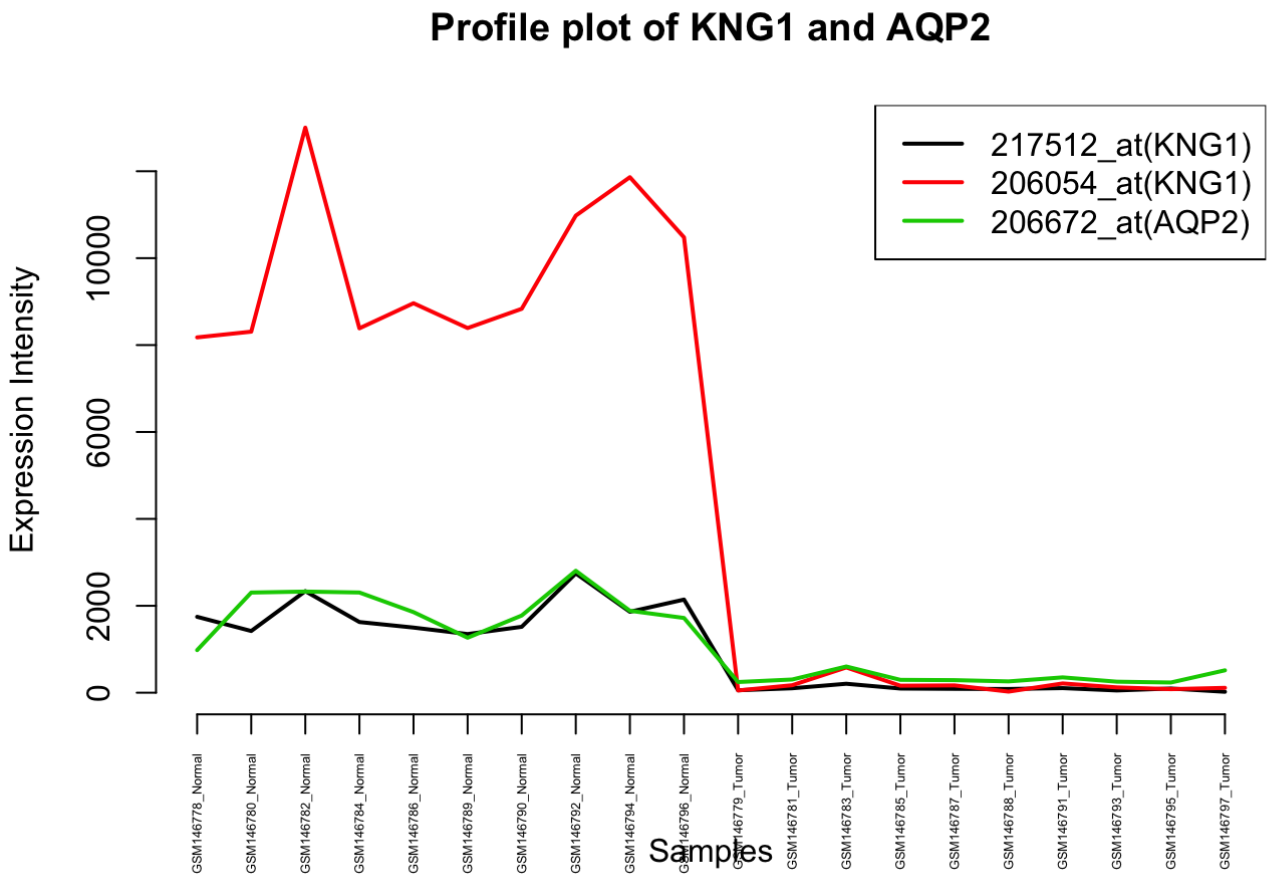
```
# I had trouble accessing NetAffx because it requires an account and for some reason, I could not register for on
e. I was unable to find information regarding Probeset IDs on genecards. Thus, I decided to just use biomaRt.
library("biomaRt")
ensembl = useMart("ENSEMBL_MART_ENSEMBL", host="www.ensembl.org")
ensembl = useMart("ENSEMBL_MART_ENSEMBL",dataset="hsapiens_gene_ensembl", host="www.ensembl.org")
probesets<-c("affy_hg_u133a","affy_hg_u133a_2")
IDs<-getBM(attributes=c(probesets,"hgnc_symbol"), filters = 'hgnc_symbol', values = c("KNG1","AQP2"), mart = ense
mbl)
print(IDs)
```

```
##   affy_hg_u133a affy_hg_u133a_2 hgnc_symbol
## 1                                      AQP2
```

```
## 2     206672_at       206672_at       AQP2
## 3     206054_at       206054_at       KNG1
## 4     217512_at       206054_at       KNG1
## 5     217512_at       217512_at       KNG1
## 6     206054_at       217512_at       KNG1
## 7                                     KNG1
```

```
# I found 2 probesets for KNG1 and 1 probeset for AQP2
KNG1<-c("217512_at","206054_at")
AQP2<-"206672_at"

#plot profile plot
target.genes<-renal[c(KNG1,AQP2),]
plot(c(1,ncol(target.genes)),range(target.genes),type='n',main="Profile plot of KNG1 and AQP2",xlab="Samples",yla
b="Expression Intensity",axes=F)
axis(side=1,at=c(1:ncol(target.genes)),labels=dimnames(target.genes)[[2]],cex.axis=0.4,las=2)
axis(side=2)
for(i in 1:nrow(target.genes)) {
dat.y <- as.numeric(target.genes[i,])
lines(c(1:ncol(target.genes)),dat.y,col=i,lwd=2)
}
legend(x="topright",legend=c("217512_at(KNG1)","206054_at(KNG1)","206672_at(AQP2)"), col=1:3,lwd=2)
```




```
# The plots of these genes seem to indicate normal renal function because there is a clear distinction between th
e expression levels of the normal samples and the expression levels of the tumour samples in these genes. The exp
ression levels of these genes are consistently higher in the normal samples compared to the tumor samples
```

9.) We want to assess the accuracy of missing value imputation. So assign the KNG1 probeset (206054_at) an NA value, only for array GSM146784. Be sure to first save the original value before replacing it with an NA. Also cast the data frame to a matrix to run this function. (2pts.)

```
original.value<-renal["206054_at","GSM146784_Normal"]
original.gene<-renal["206054_at",]
```

```
renal["206054_at","GSM146784_Normal"]<-NA
renal<-as.matrix(renal)
```

10.) Now estimate the missing values in the array using 6 nearest neighbors and Euclidean distance with the impute.knn() function. (2pts.)

```
library(impute)
renal.imputed<-impute.knn(renal,k=6)
```

11.) Look at the value that was imputed for your gene and calculate the relative error of this value using the actual value that you saved. (2pts.)

```
predicted.value<-renal.imputed$data["206054_at","GSM146784_Normal"]
print(predicted.value)
```

```
## [1] 7559.533
```

```
relative.error<-abs(original.value-predicted.value)/abs(original.value)
relative.error
```

```
## [1] 0.09847789
```

12.) Now impute the missing values using the SVD imputation method. This is in the pcaMethods package and the function is called pca ()with method svdImpute and set nPcs=9. To retrive the output matrix, see the help file. (2pts.)

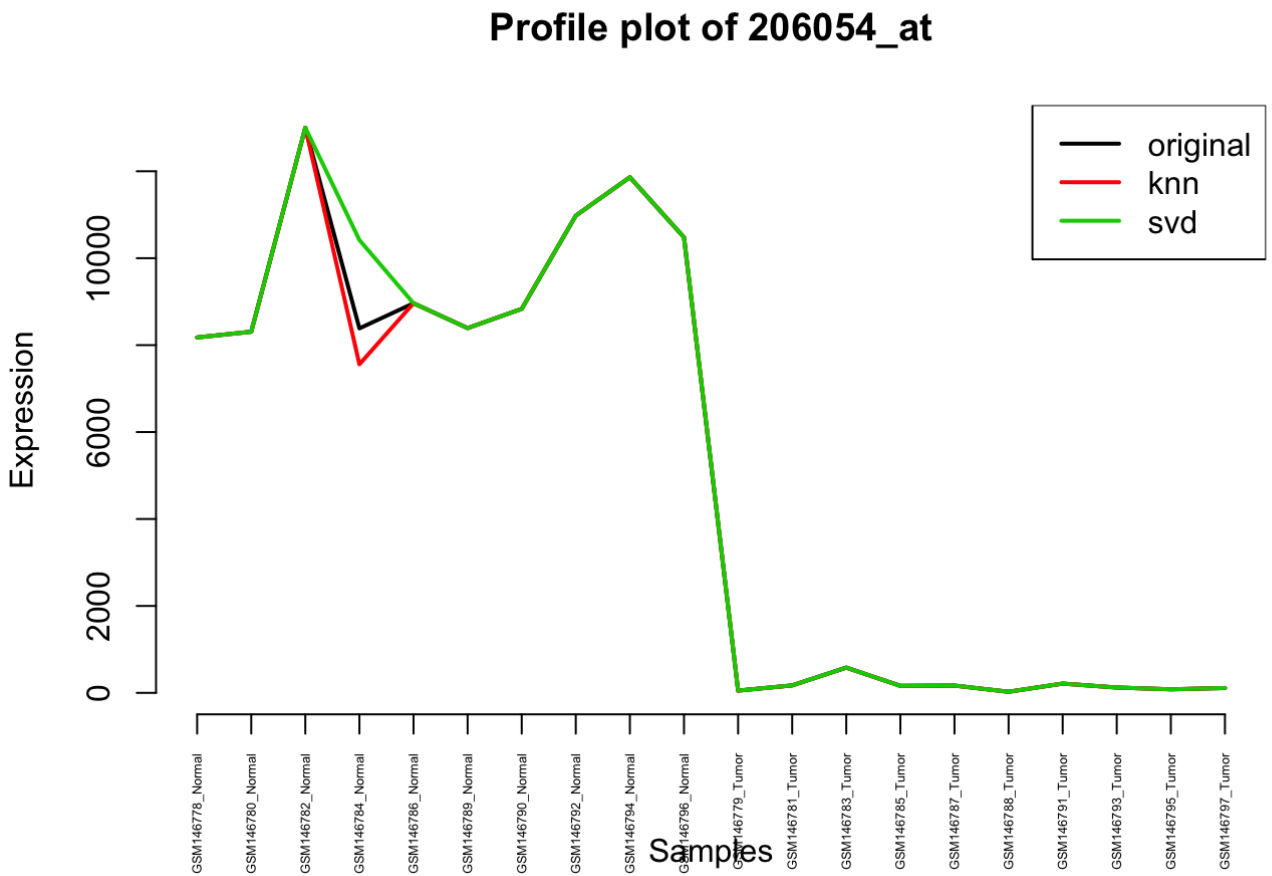
```
library(pcaMethods)
renal.imputed.2<-pca(renal,method="svdImpute", nPcs=9)
svd<-completeObs(renal.imputed.2)
print(svd["206054_at","GSM146784_Normal"])
```

```
## [1] 10418
```

13.) Finally, plot a gene profile plot of the probeset for this gene, where the two different imputed values are represented as different colored points and the actual value is a third point. (6pts.)

```
results<-list(original=as.numeric(original.gene),knn=as.numeric(renal.imputed$data["206054_at",]),svd=as.numeric
(svd["206054_at",]))
plot(c(1,ncol(original.gene)),range(original.gene),type='n',main="Profile plot of 206054_at",xlab="Samples",ylab=
"Expression",axes=F)

axis(side=1,at=c(1:ncol(original.gene)),labels=dimnames(original.gene)[[2]],cex.axis=0.4,las=2)
axis(side=2)
for(i in 1:length(results)) {
dat.y <- results[[i]]
lines(c(1:ncol(original.gene)),dat.y,col=i,lwd=2)
}
legend(x="topright",legend=c("original","knn","svd"),col=1:length(results),lwd=2)
```



Generate the code and plots for each. Turn in the visuals, code, and an explanation of the questions asked. Paste all information into a PDF doc.