LAB 8

STAT 28

March 15, 2017

In this lab, we will discuss the heatmap visualization technique.

Heatmap Lecture Review

We will read in the data on gene expression from breast tumors.

```
breast<-read.csv("highVarBreast.csv")
```

We can visualize the correlations of our variables. We will not use the `corrgram` function, however; that function is only appropriate for a small number of variables. We will instead create a heatmap using a heatmap function. There are many heatmap functions in R. The default `heatmap` comes with the standard R, but is not very good. Better options are `heatmap.2` in `gplots`; `heatmap.plus` in the `heatmap.plus` package; and `aheatmap` in the `NMF` package. I favor the `pheatmap` function in the `pheatmap` package. It's got a couple of strange quirks, but generally gives the nicest pictures, I find.

```
library(pheatmap)
```

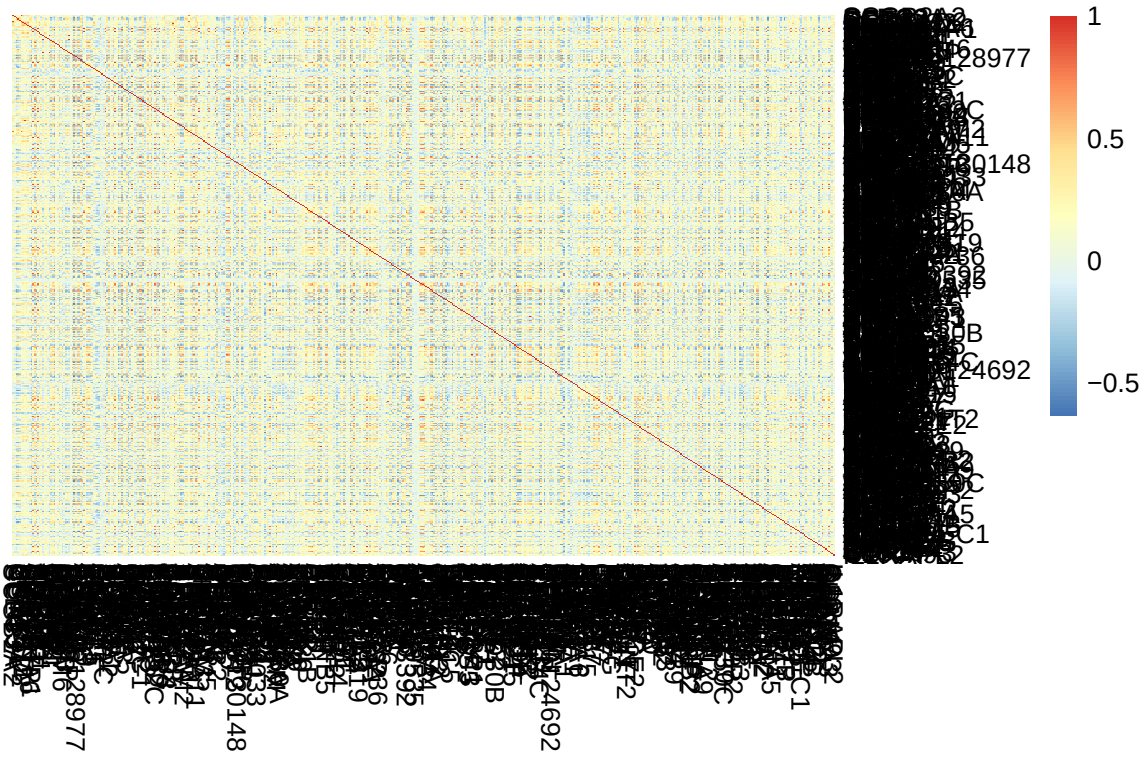
```
## Warning: package 'pheatmap' was built under R version 3.4.4
```

Now to plotting the correlations. First we are going to calculate all the correlations with the `cor` function. This function calculates the correlations among the *columns* of a matrix. Notice, we need to remove the non-continuous variables.

```
corMat<-cor(breast[,-c(1:7)])
```

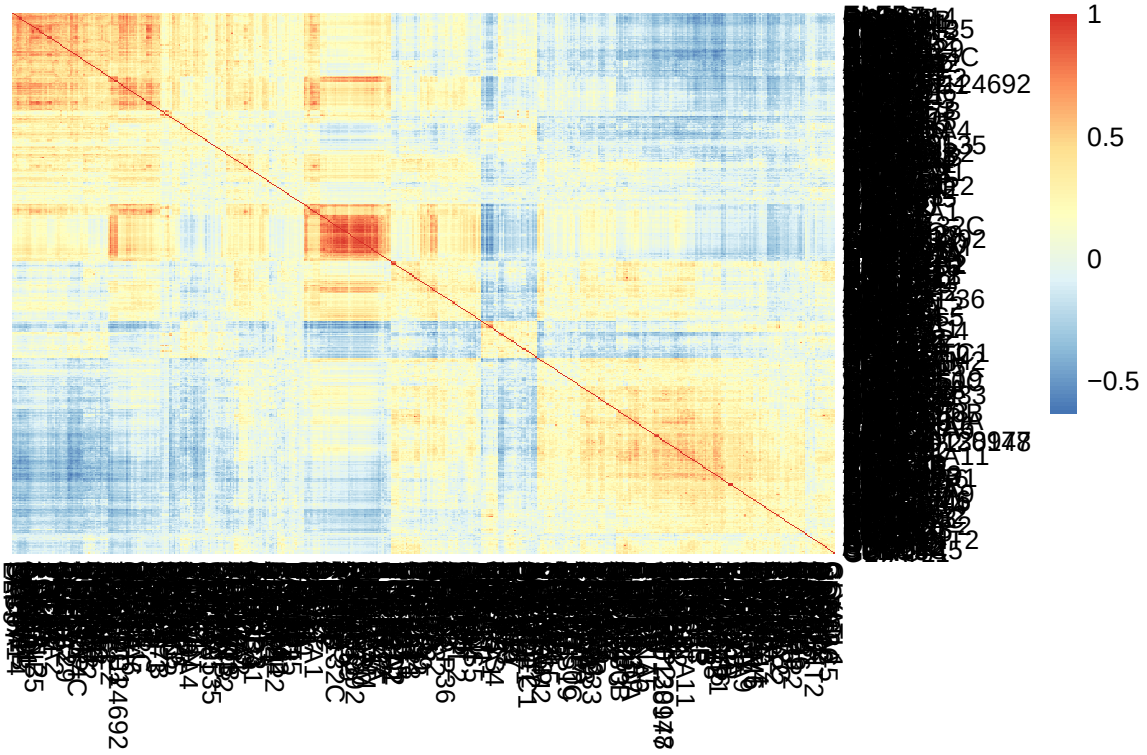
Now we run `pheatmap`. I have supressed ordering of the samples by turning off both column and row clustering (`cluster_rows = FALSE`, and `cluster_cols = FALSE`). This was for teaching purposes during class, to show what it would look like without clustering, but not what you would want to actually do.

```
pheatmap(corMat,cluster_rows = FALSE, cluster_cols = FALSE )
```



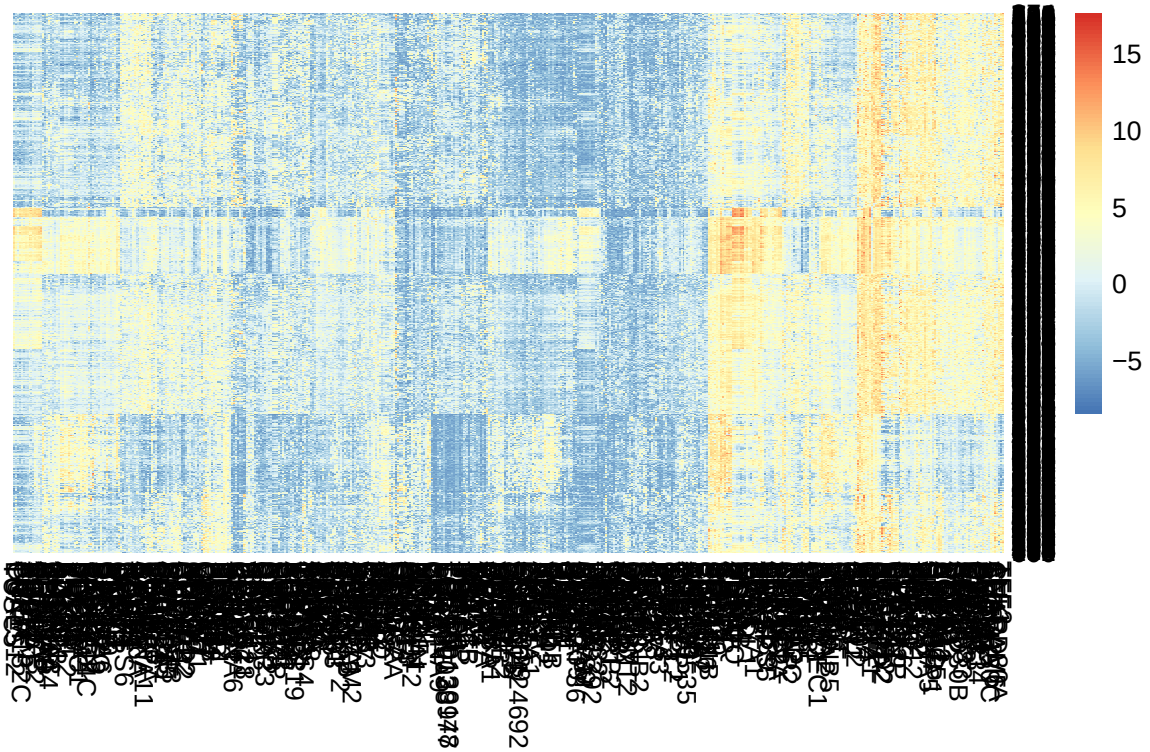
Now we will draw the same heatmap with the rows and samples clustered. However, I chose not to show the hierarchical clustering tree by setting `treeheight_row =0` and `treeheight_col = 0`. Again this was for teaching purposes, but this is actually sometimes useful if you don't want the clutter of the dendrogram. You'll see this same code a good deal, because I don't actually get to the idea of clustering until later.

```
pheatmap(corMat,cluster_rows = TRUE, cluster_cols = TRUE, treeheight_row =0, treeheight_col = 0)
```



Now we do a heatmap of the actual data, rather than the correlation.

```
pheatmap(breast[,-c(1:7)],cluster_rows = TRUE, cluster_cols = TRUE, treeheight_row =0, treeheight_col =
```



Now we are going to demonstrate some options with `pheatmap` to show how you can customize it. First, we want to draw colors next to the samples to identify different characteristics of the samples (e.g. normal or cancer samples). So I am going to create vectors of colors, and then give names to that vector that *exactly* match the levels of the variable. Notice that I am **NOT** making the vector of colors the same length as the factor (which is what I would do in say `plot`). `pheatmap` will handle that internally. I will do this for three variables, `TypeSample`, `EstReceptor`, and `Progesteron`.

```
typeCol<-c("red","black","yellow")
names(typeCol)<-levels(breast$TypeSample)
estCol<-palette()[c(4,3,5)]
names(estCol)<-levels(breast$EstReceptor)
proCol<-palette()[5:7]
names(proCol)<-levels(breast$Progesteron)
```

The last change I'm going to make is to make my colors "max out" after a certain point, so that all values greater than a certain amount get the same value. This way, I don't "waste" a lot of colors trying to show the value of outliers. I will choose the 1% and 99% quantiles of the data.

```
qnt<-quantile(as.numeric(data.matrix((breast[,-c(1:7)]))),c(0.01,.99))
```

Now I make a vector of break points of length 50. These will be my bins.

```
brks<-seq(qnt[1],qnt[2],length=20)
head(brks)
```

```
## [1] -5.744770 -4.949516 -4.154261 -3.359006 -2.563751 -1.768496
```

Now I'm going to change the color scheme for the heatmap by giving a smooth gradient of colors to `pheatmap`, using the `colorRampPalette` function which will create a vector of colors that smoothly interpolate from a

list of colors you give it. This is a color scheme I made up that I use frequently and like.

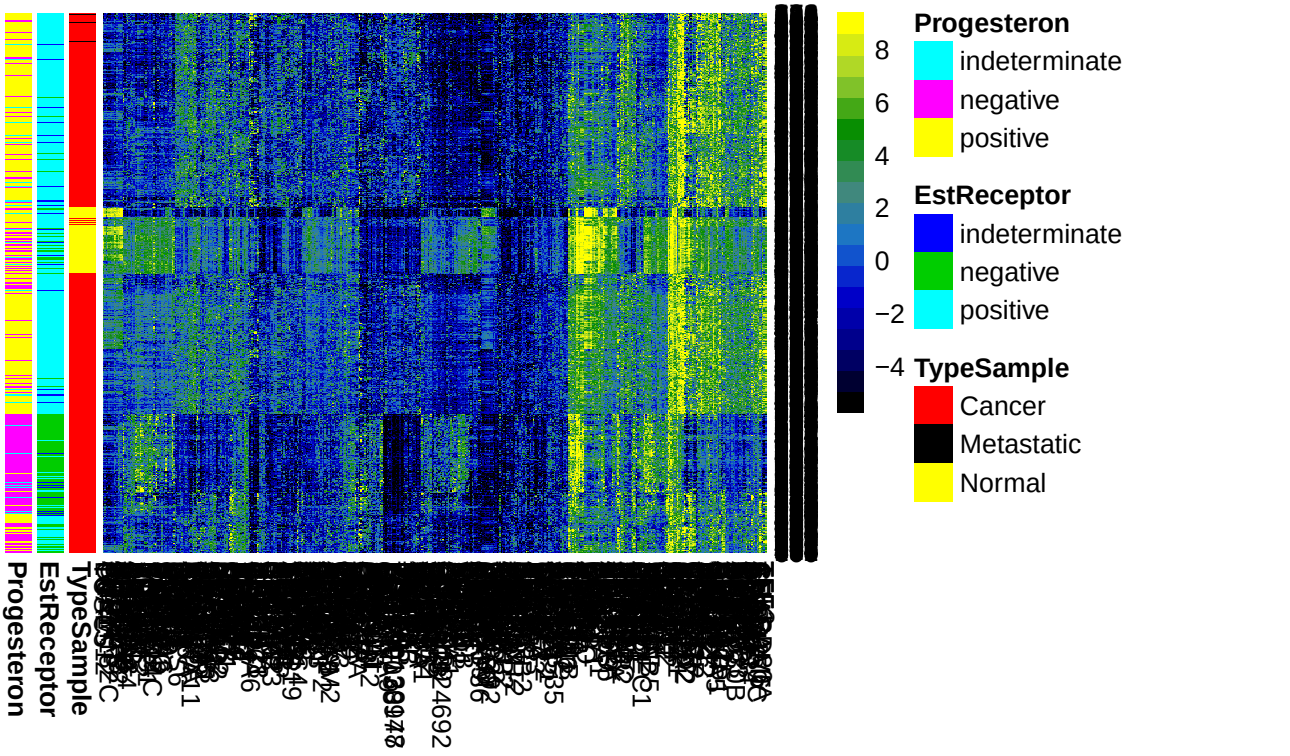
```
seqPal5<-colorRampPalette(c("black","navyblue","mediumblue","dodgerblue3","aquamarine4","green4","yellow
```

I'm almost set, only I have to decide what is the annotation I want to show. I want it to be the columns saved in the data that are "TypeSample" (normal/cancer), "EstReceptor" (the estrogen receptor status), and "Progesteron" (the progesteron receptor status). One quirk of `pheatmap` to do this labelling, you must have row names for both your data matrix and the annotation matrix that match, so we are going to give rownames to our breast dataset (that are just numbers!)

```
row.names(breast)<-c(1:nrow(breast))
```

Now I draw the heatmap with all of these features. `annotation_row` is the variable(s) that define the colors I want to draw next to the samples/rows (the argument `annotation_col` does the same thing for columns). `annotation_colors` is a list that gives the colors for each of these variables. Note that the name of the list exactly match the name of the variable in the `annotation_row`. Note that if you don't give colors, `pheatmap` will automatically create some.

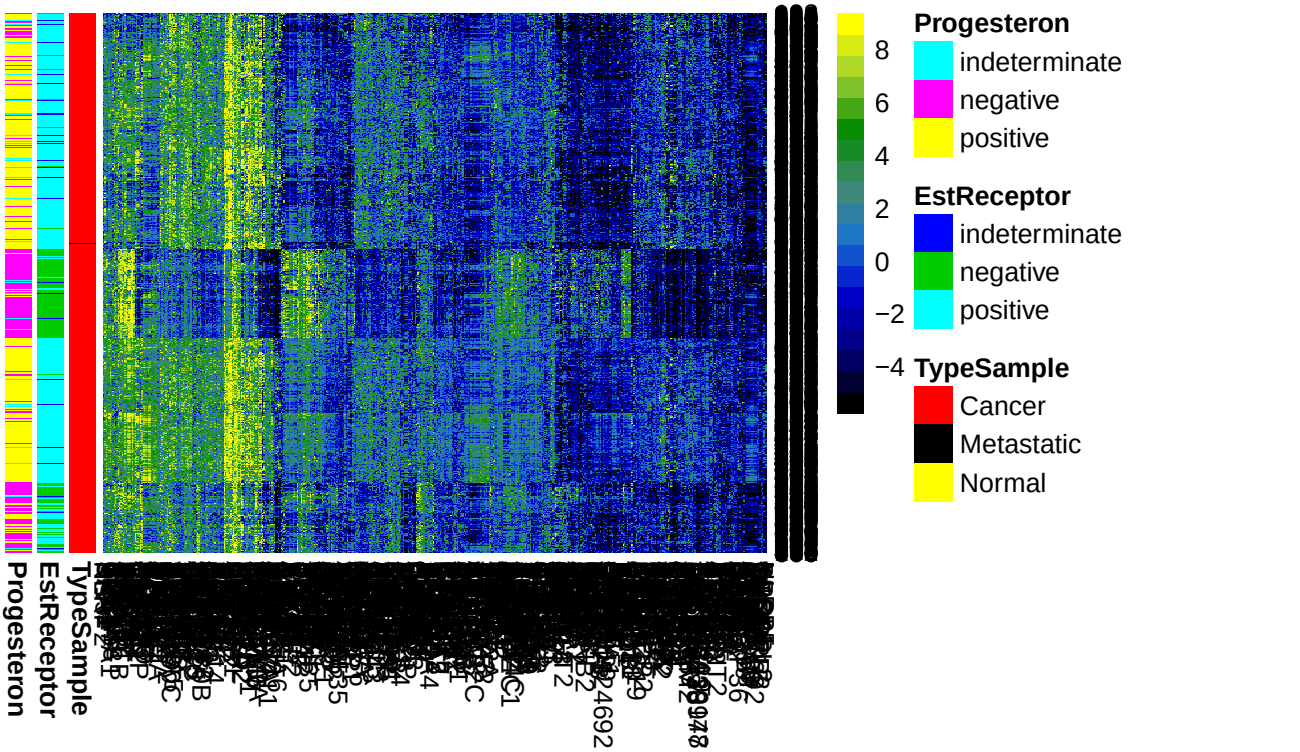
```
fullHeat<-pheatmap(breast[,-c(1:7)],cluster_rows = TRUE, cluster_cols = TRUE,
    treeheight_row =0, treeheight_col = 0,
    color=seqPal5,
    breaks=brks,
    annotation_row=breast[,5:7],
    annotation_colors = list("TypeSample"=typeCol,"EstReceptor"=estCol,"Progesteron"=proCol))
```



Now we will draw a heatmap for only the cancer samples. Notice I don't have to change much of the setup from above.

```
whCancer<-which(breast$Type!="Normal")
pheatmap(breast[whCancer,-c(1:7)],cluster_rows = TRUE, cluster_cols = TRUE,
    treeheight_row =0, treeheight_col = 0,
    color=seqPal5,
    breaks=brks,
```

```
    annotation_row=breast[whCancer,5:7],
    annotation_colors = list("TypeSample"=typeCol,"EstReceptor"=estCol,"Progesteron"=proCol))
```



Here we are going to "center" the data, by subtracing out the median of each variable so that they are on a closer scale. I am actually going to center the data in two ways. First by subracting off the mean and then the median, just to show you a few more functions.

There is a simple function for subtracting off the mean, called `scale`. This function subtracts off the mean of each variable (column) and returns a data.frame that has been centered. There is also an option to divide each variable by its standard deviation to put them on the same scale (`scale=TRUE` would do that).

```
breastCenteredMean<-scale(breast[,-c(1:7)],center=TRUE,scale=FALSE)
```

To subtract off the median requires a bit more work. First we need to calculate the median of each variable, which we will do with 'apply. This will return a vector of the length of the number of variables.

```
colMedian<-apply(breast[,-c(1:7)],2,median)
```

Then we need to subtract them from every column. There is a function `sweep` that will take a vector and subtract it from every column in the matrix. It works similarly to `apply`. The first argument is the matrix/data.frame; the second argument, like apply, is `MARGIN` that defines whether you are subtracting from the rows (1) or columns (2). Then you give the vector that you want to subtract, and finally the operation you want to perform ("-","+","/","*" are common ones though you could write your own).

```
breastCenteredMed<-sweep(breast[,-c(1:7)],MARGIN=2, colMedian,"-")
```

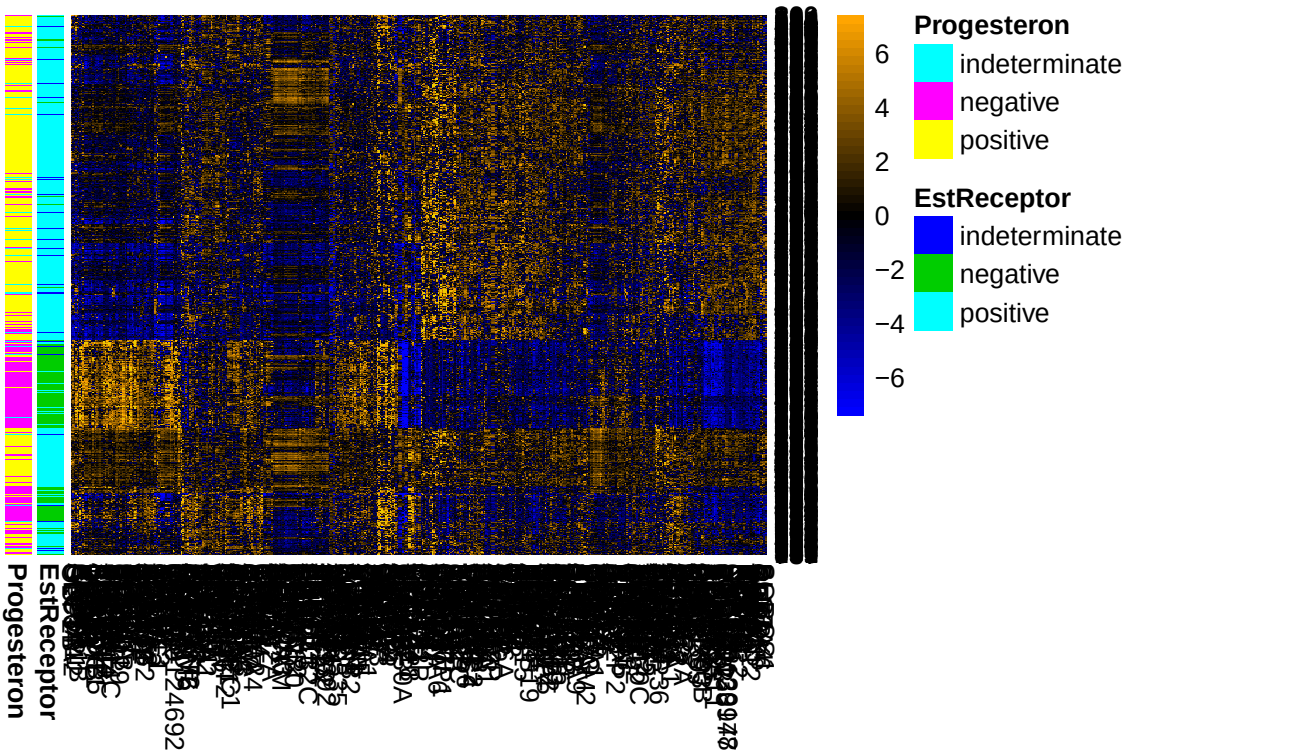
If this feels like you could have just done this with a call to `apply`, you are right! This is just such a common task that they have a helper function. In particular, it guarantees that the output will be the same dimension as the input, which is frequently not the case for `apply` (which sometimes winds up "flipping" the rows and columns on you.)

Now we are going to create breaks for this centered data that are symmetric around zero, and also chop off the really outlying points.

```
qnt<-max(abs(quantile(as.numeric(data.matrix((breastCenteredMed[,-c(1:7)]))),c(0.01,.99))))
brksCentered<-seq(-qnt,qnt,length=50)
```

We are also going to create a different color scale. Here's another one that I like for centered data.

```
seqPal2<- colorRampPalette(c("orange","black","blue"))(length(brksCentered)-1)
seqPal2<-(c("yellow","gold2",seqPal2))
seqPal2<-rev(seqPal2)
pheatmap(breastCenteredMed[whCancer,-c(1:7)],cluster_rows = TRUE, cluster_cols = TRUE,
    treeheight_row =0, treeheight_col = 0,
    color=seqPal2,
    breaks=brksCentered,
    annotation_row=breast[whCancer,6:7],
    annotation_colors = list("TypeSample"=typeCol,"EstReceptor"=estCol,"Progesteron"=proCol))
```

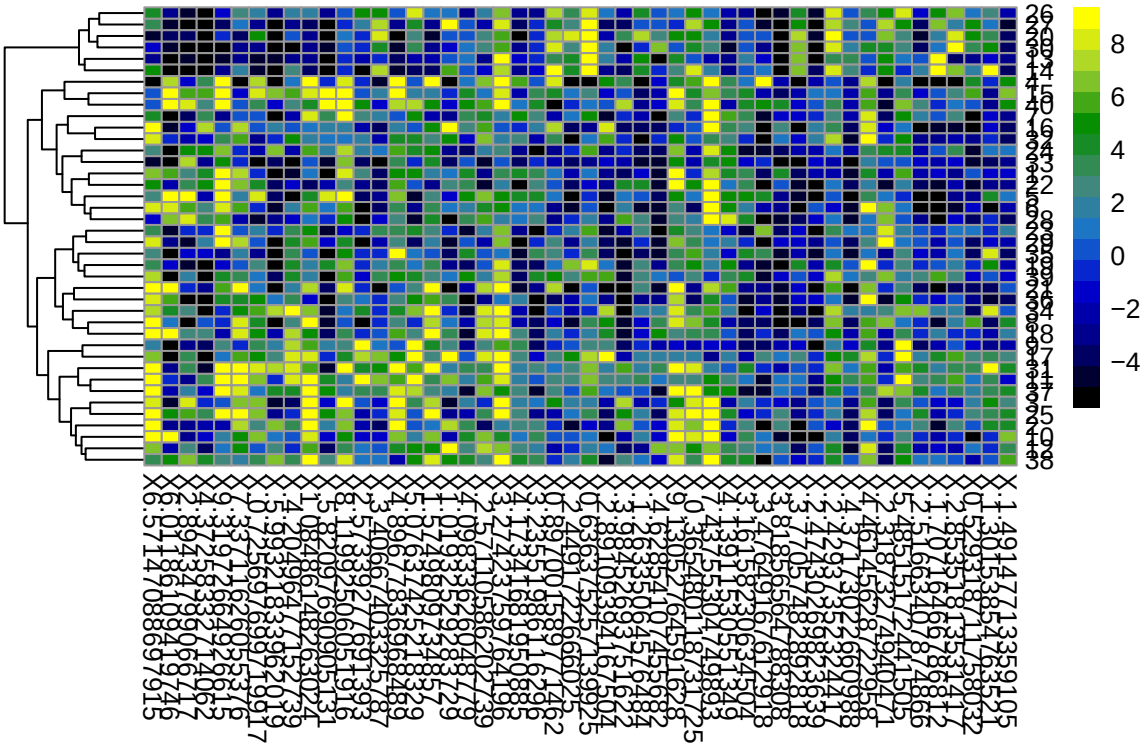


Now we are going to work with a small subset of genes and samples so that we can more clearly see the dendrograms from the clustering. By not setting `treeheight_row`, it gives the dendrogram for the rows (but both rows and columns are clustered).

```
smallBreast<-read.csv("smallVarBreast.csv" ,header=TRUE)
row.names(smallBreast)<-1:nrow(smallBreast)
```

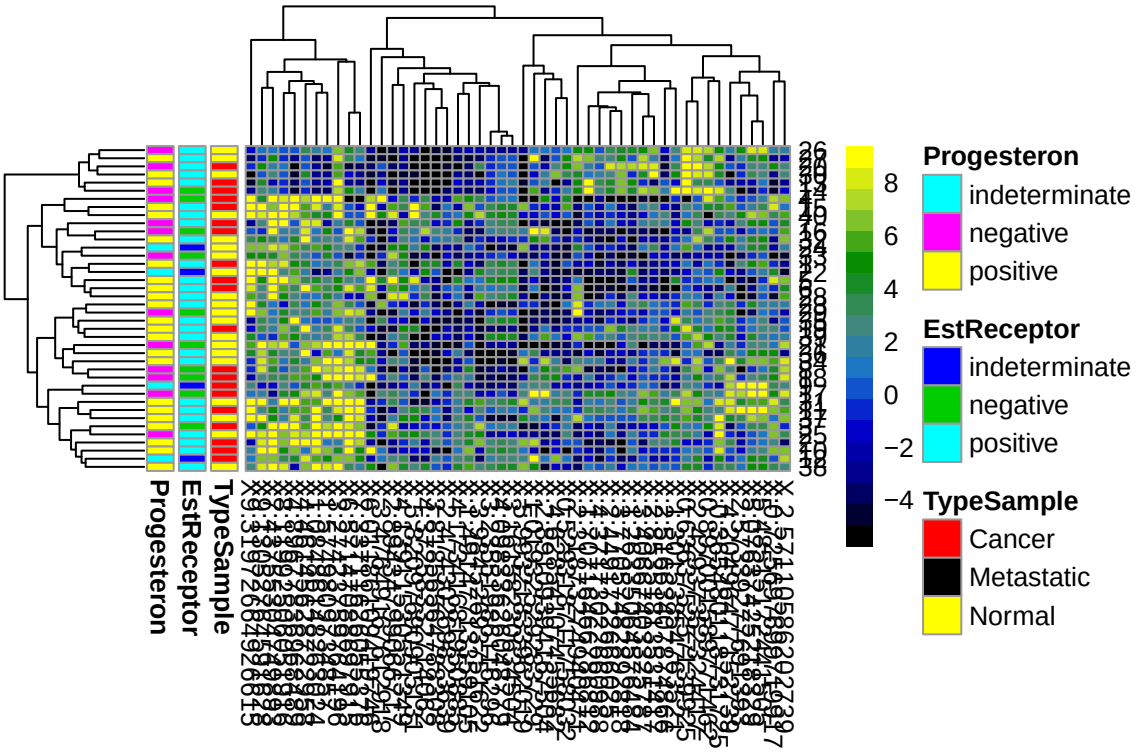
The following command shows the dendrogram, but only for the rows:

```
pheatmap(smallBreast[,-c(1:7)],cluster_rows = TRUE, cluster_cols = FALSE,  treeheight_col = 0,
    breaks=brks,col=seqPal5)
```



The following command shows the dendrogram for both:

```
pheatmap(smallBreast[,-c(1:7)],,cluster_rows = TRUE, cluster_cols = TRUE,
    breaks=brks,col=seqPal5,annotation_row=smallBreast[,5:7],
    annotation_colors=list("TypeSample"=typeCol,"EstReceptor"=estCol,"Progesteron"=proCol))
```



Exercises

Place rated dataset

The data is taken from a book named **Places Rated Almanac**. They used nine rating criteria to rate cities. For all but two of the criteria, the higher the score, the better. For Housing and Crime, the lower the score the better. The scores are computed using the following component statistics for each criterion:

- Climate & Terrain: very hot and very cold months, seasonal temperature variation, heating- and cooling-degree days, freezing days, zero-degree days, ninety-degree days.
- Housing: utility bills, property taxes, mortgage payments.
- Health Care & Environment: per capita physicians, teaching hospitals, medical schools, cardiac rehabilitation centers, comprehensive cancer treatment centers, hospices, insurance/hospitalization costs index, flouridation of drinking water, air pollution.
- Crime: violent crime rate, property crime rate.
- Transportation: daily commute, public transportation, Interstate highways, air service, passenger rail service.
- Education: pupil/teacher ratio in the public K-12 system, effort index in K-12, accademic options in higher education.
- The Arts: museums, fine arts and public radio stations, public television stations, universities offering a degree or degrees in the arts, symphony orchestras, theatres, opera companies, dance companies, public libraries.
- Recreation: good restaurants, public golf courses, certified lanes for tenpin bowling, movie theatres, zoos, aquariums, family theme parks, sanctioned automobile race tracks, pari-mutuel betting attractions, major- and minor- league professional sports teams, NCAA Division I football and basketball teams, miles of ocean or Great Lakes coastline, inland water, national forests, national parks, or national wildlife refuges, Consolidated Metropolitan Statistical Area access.
- Economics: average household income adjusted for taxes and living costs, income growth, job growth.

Read Data.

```
place_rated <- read.csv("place.csv", stringsAsFactors = FALSE)
place_rated[, 1:9] <- scale(place_rated[, 1:9])
CA_NY_cities <- which(place_rated$state %in% c("CA", "NY"))
example_cities <- CA_NY_cities[c(10, 13, 15, 22, 24, 25, 26)]
row.names(place_rated) <- paste0(place_rated$city, place_rated$state)
```

Exercise 1

We learned two types of heatmaps in lecture 4. One is heatmap for correlation matrix, the other is heatmap for data matrix.

Cor function

```
cor(1:10, 1:10)
```

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

```
mat = matrix(rnorm(100), 10, 10)
cor(mat)
```

```
##              [,1]       [,2]        [,3]        [,4]        [,5]
##  [1,]  1.00000000 -0.3748896  0.24491604  0.08844734 -0.62207958
##  [2,] -0.37488963  1.0000000  0.31638618 -0.11300504  0.69633908
##  [3,]  0.24491604  0.3163862  1.00000000  0.30240342  0.01649312
##  [4,]  0.08844734 -0.1130050  0.30240342  1.00000000 -0.20771026
##  [5,] -0.62207958  0.6963391  0.01649312 -0.20771026  1.00000000
##  [6,] -0.06981201  0.2723582 -0.20452417  0.56498275  0.17626651
```

```
##  [7,]  0.35774739  0.0194732  0.02428961 -0.39919096 -0.21767497
##  [8,]  0.18180901 -0.6304495 -0.22811855 -0.31807418 -0.10494261
##  [9,]  0.16722708  0.3557131  0.42662037  0.09684287  0.24430231
## [10,]  0.24894138  0.5335257  0.40124583 -0.26563911  0.47850926
##               [,6]        [,7]        [,8]        [,9]        [,10]
##  [1,] -0.069812007  0.35774739  0.18180901  0.16722708  0.248941376
##  [2,]  0.272358237  0.01947320 -0.63044950  0.35571311  0.533525747
##  [3,] -0.204524166  0.02428961 -0.22811855  0.42662037  0.401245835
##  [4,]  0.564982752 -0.39919096 -0.31807418  0.09684287 -0.265639107
##  [5,]  0.176266515 -0.21767497 -0.10494261  0.24430231  0.478509256
##  [6,]  1.000000000 -0.45385486 -0.27350872 -0.12422555 -0.003595349
##  [7,] -0.453854859  1.00000000 -0.25836332  0.08975339  0.358744362
##  [8,] -0.273508725 -0.25836332  1.00000000 -0.37751465 -0.047446673
##  [9,] -0.124225554  0.08975339 -0.37751465  1.00000000  0.414739710
## [10,] -0.003595349  0.35874436 -0.04744667  0.41473971  1.000000000
```

```
cor(mat[, 1:5])
```

```
##             [,1]       [,2]       [,3]        [,4]        [,5]
## [1,]  1.00000000 -0.3748896 0.24491604  0.08844734 -0.62207958
## [2,] -0.37488963  1.0000000 0.31638618 -0.11300504  0.69633908
## [3,]  0.24491604  0.3163862 1.00000000  0.30240342  0.01649312
## [4,]  0.08844734 -0.1130050 0.30240342  1.00000000 -0.20771026
## [5,] -0.62207958  0.6963391 0.01649312 -0.20771026  1.00000000
```

```
cor(mat[,1:3], mat[,4:5])
```

```
##             [,1]        [,2]
## [1,]  0.08844734 -0.62207958
## [2,] -0.11300504  0.69633908
## [3,]  0.30240342  0.01649312
```

(a) Calculate the correlation matrix between nine rating criteria using function `cor`. What is the correlation between arts and education?

```
# insert your code here and save the correlation matrix as
# `cor_place_rated` and call this matrix.

# cor_place_rated <-
# cor_place_rated
```

(b) Plot the heatmap for the correlation matrix using the `pheatmap` function from `pheatmap` package. If you are going to divide the nine rating criteria into two categories based on similarity, how would you split the variables?

```
# insert your code here for heatmap
library(pheatmap)
```

< write answer here >

(c) Plot the heatmap for data matrix using the `pheatmap` function.

```
# insert your code here for the heatmap
```