Assignment 1: STAT5900 F

by

Eralda Gjika

(Student ID: 101248793)

Statistical Methods for Big Genomic Data

(Due: 11:59 pm Thursday Oct 13, 2022)

School of Mathematics and Statistics
Carleton University
13 October 2022

Contents

1 Question 1

Conduct differential expression analysis to find genes that are differentially expressed between cancer and healthy patients using *limma*. Don't forget to check the distributional assumption. Note that you need to make you do not want to miss out the signal, so use an appropriate multiple testing correction. Summarize your findings and create a heatmap of all the differentially expressed genes

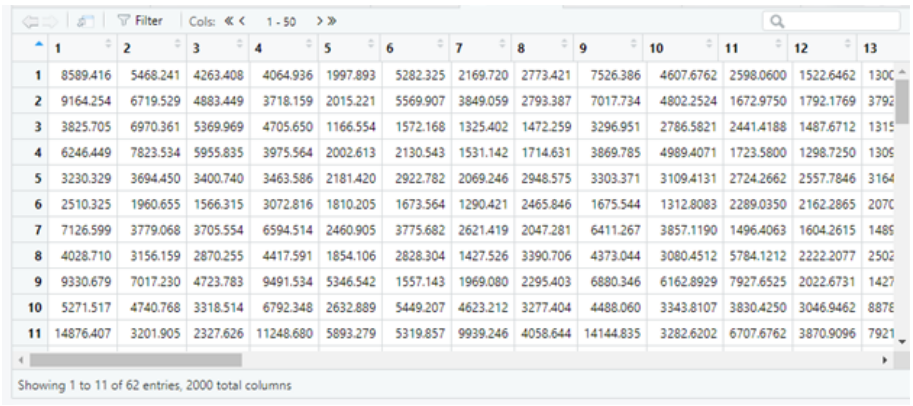
1.1 Introduction with the data

The dimension of our data are 62 rows and 2000 columns. Which means we have 2000 genes and 62 samples. From which 22 samples of class 1 which corresponds to normal tissues and 40 samples of class 2 which corresponds to tumor tissues. (Source of data: http://microarray.princeton.edu/oncology/affydata/index.html) The dataset consist of three main variables:

X a (62 x 2000) matrix giving the expression levels of 2000 genes for the 62 Colon tissue samples. Each row corresponds to a patient, each column to a gene.

Y a numeric vector of length 62 giving the type of tissue sample (tumor or normal).

gene.names a vector containing the names of the 2000 genes for the gene expression matrix X.


Filter Cols: « < 1 - 50 > »

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	8589.416	5468.241	4263.408	4064.936	1997.893	5282.325	2169.720	2773.421	7526.386	4607.6762	2598.0600	1522.6462	1300
2	9164.254	6719.529	4883.449	3718.159	2015.221	5569.907	3849.059	2793.387	7017.734	4802.2524	1672.9750	1792.1769	3792
3	3825.705	6970.361	5369.969	4705.650	1166.554	1572.168	1325.402	1472.259	3296.951	2786.5821	2441.4188	1487.6712	1315
4	6246.449	7823.534	5955.835	3975.564	2002.613	2130.543	1531.142	1714.631	3869.785	4989.4071	1723.5800	1298.7250	1309
5	3230.329	3694.450	3400.740	3463.586	2181.420	2922.782	2069.246	2948.575	3303.371	3109.4131	2724.2662	2557.7846	3164
6	2510.325	1960.655	1566.315	3072.816	1810.205	1673.564	1290.421	2465.846	1675.544	1312.8083	2289.0350	2162.2865	2070
7	7126.599	3779.068	3705.554	6594.514	2460.905	3775.682	2621.419	2047.281	6411.267	3857.1190	1496.4063	1604.2615	1489
8	4028.710	3156.159	2870.255	4417.591	1854.106	2828.304	1427.526	3390.706	4373.044	3080.4512	5784.1212	2222.2077	2502
9	9330.679	7017.230	4723.783	9491.534	5346.542	1557.143	1969.080	2295.403	6880.346	6162.8929	7927.6525	2022.6731	1427
10	5271.517	4740.768	3318.514	6792.348	2632.889	5449.207	4623.212	3277.404	4488.060	3343.8107	3830.4250	3046.9462	8878
11	14876.407	3201.905	2327.626	11248.680	5893.279	5319.857	9939.246	4058.644	14144.835	3282.6202	6707.6762	3870.9096	7921

Showing 1 to 11 of 62 entries, 2000 total columns

Figure 1: Screenshot of the data Colon$X

1.2 Pre-Processing the data Colon X

At this step we will try to have a better understanding of the data. Using histograms, Boxplot, Q-Q plot and other graphical test we will observe the distribution of our data and decide if any transformation (normalization) should be used to improve further analysis. Figure 2 shows the histogram for the first 4 genes. We do not observe a clear normal distribution but we may use other graphical visualization to decide.

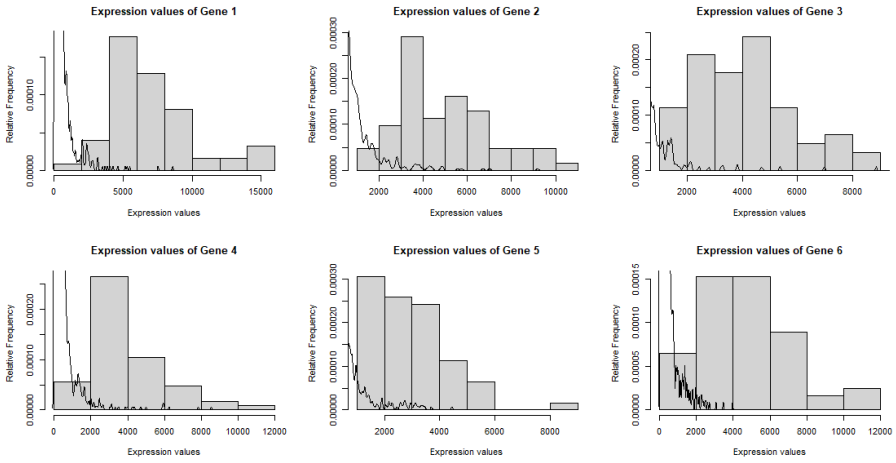


Figure 2: Histogram for the first 6 genes

Observing the boxplot of the first 70 genes we obtain Figure 3. We clearly observe a distribution which is not normal for many of the genes. We may suggest a normalization of the data based on the distributions showed from boxplot graphs.

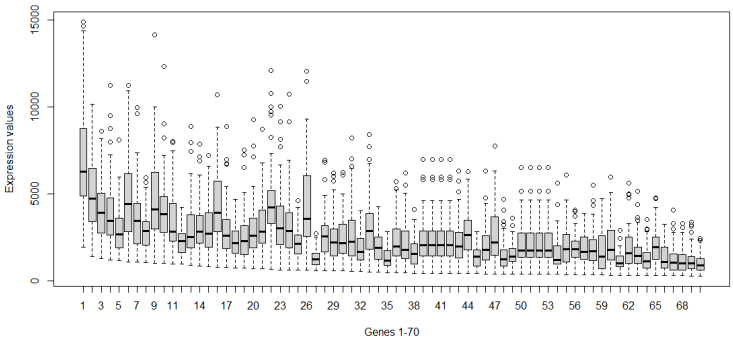


Figure 3: Boxplot for the first 70 genes

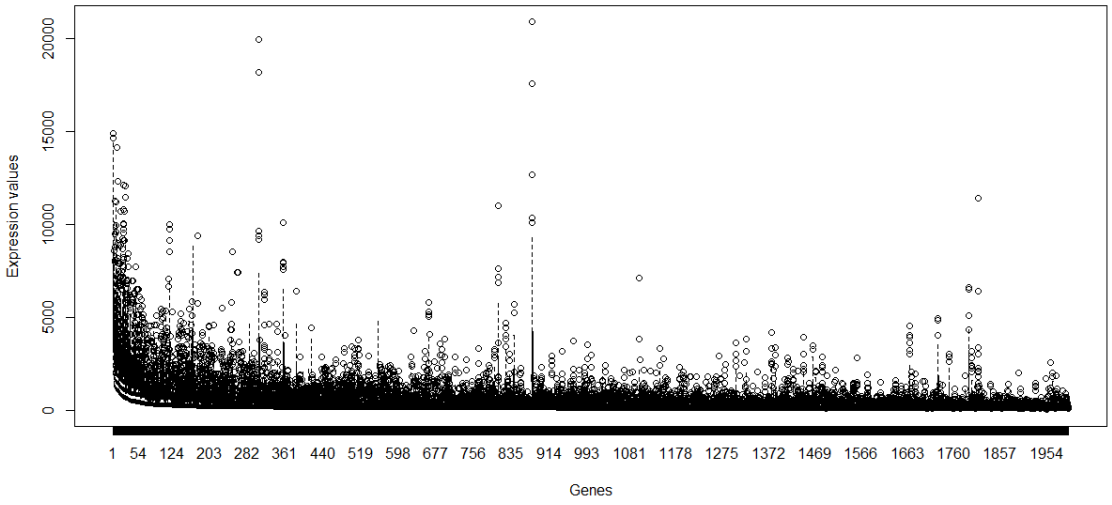


Figure 4: Boxplot for all genes

For a fast overview of the distribution of the data let's observe the density plot as below (Figure 5). When we increase the number of genes from 20 up to 2000 we observe a change in the distribution. Which seem to be right skewed distributed (fat tail or heavy tail distributions are a special probability distribution that exhibits a large skewness or kurtosis).

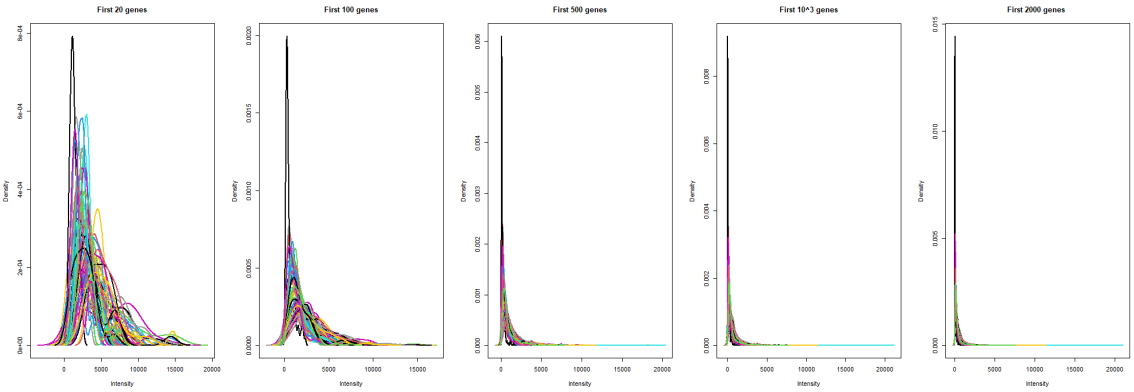


Figure 5: Density plot for different number of genes

The first genes seem to be different from the other genes (observe gene 1 which is obvious

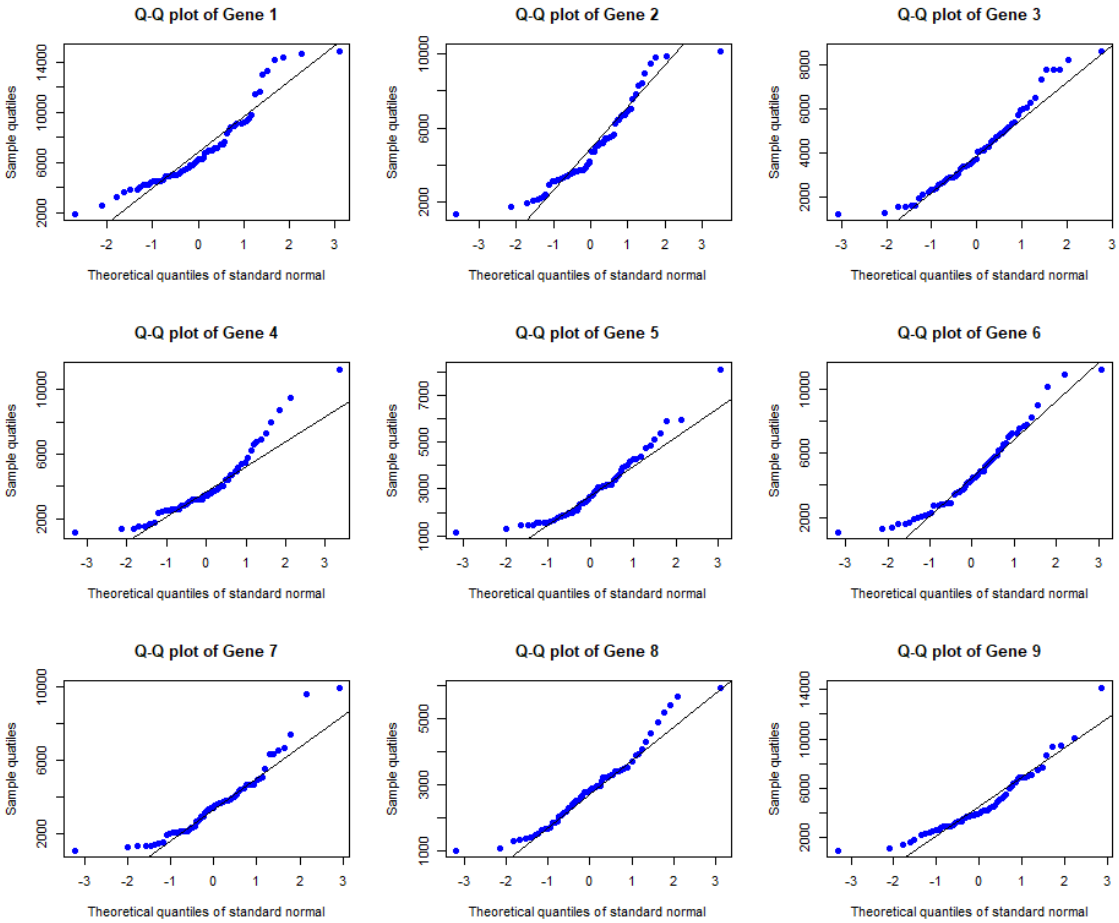


Figure 6: QQ plot for the first 9 genes

different from the others (box plot and also density plot shows it)). So, again here we confirm the necessity of transforming the data.

Also from the QQplot (Figure 6) we observe some deviations at the tail of the graph which suggest outliers and so not normally distributed data. For example gene 1 and Gene 4 have visible deviations from the theoretical quantiles and so they may not be normally distributed. (Figure 4)

1.3 Transformation and distributional assumption

1.3.1 First approach: $getNormMatrix()$

In the first approach I tried to normalize the data by using the reference package $getNormMatrix()$. Based from the review of this function in R it is mentioned that normalization without evaluation, TU is recommended for the normalization of gene expression data, as it has been already ranked as the best method for both scRNA-seq and bulk RNA-seq data. (**Reference:** https://www.frontiersin.org/articles/10.3389/fgene.2019.00400/full) The results for this normalization procedure in our data does not show significant improvements in the distribution of the data.

1.3.2 Second approach: $log2()$

In the second approach I used logarithm of base 2 to transform the data. And based on the visualization it looks like the second approach performed better than the first one.

1.3.3 Third approach: $ZScore$

I have also tried the $Z - score$ transformation on the data ($Z = (X - mean)/sd$). But again this process was not successful and did not overpass $log2$ transformation.

Below are a set of graphs showing how these proposed transformation performed and how the data look like after the transformation.

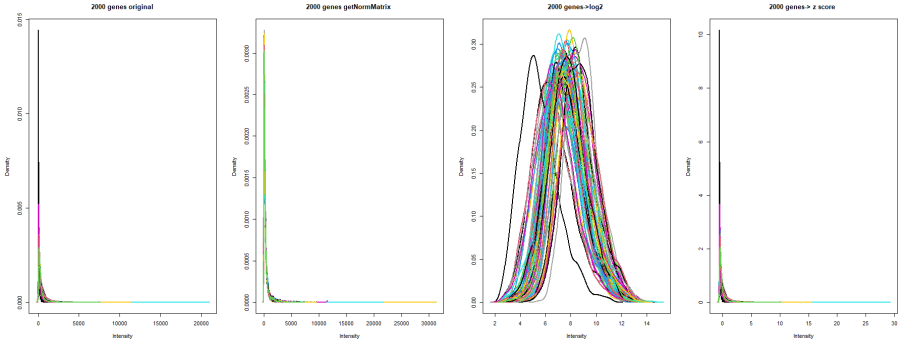


Figure 7: density plot comparison of transformation

Figure 7 gives (from left to right) the density plot for the original data, $getNormMatrix$, $log2$ and $Zscore$ transformation. Again here, we observe that $log2$ transformation offers a

more clear distribution close to normal distribution of the data. (pay attention to gene 1 it also remains a possible outliers , or an observation which may need more attention)

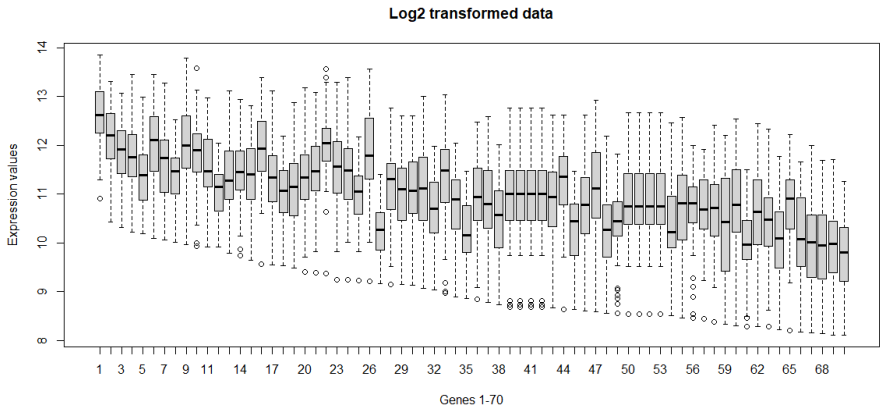


Figure 8: Boxplot of 70 genes for *log*2 transformation

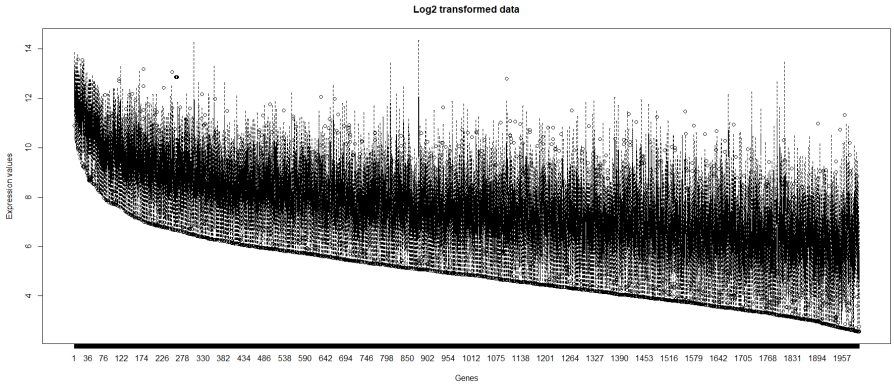


Figure 9: Boxplot of all genes for *log*2 transformation

Figure 8 and Figure 9 visualize the boxplot for *log*2 transformation approach. Observing all genes we clearly see different behaviour of the first genes (numbered as in Colon$X) and the last genes. This difference is clear and will be present when we will discuss in the other steps of gene analysis. **Here we may ask: do we have clear clusters of genes which will help us identify and explain this behavior among genes?**

(**NOTE:** in R markdown file more graphs are offered for a better display and check of the normality assumption of the three approaches.)

1.4 Multiple testing correction methods

Getting a list of deferentially expressed genes means that we need to choose an absolute threshold for the $log2$ fold change (column log2FoldChange in the output of the functions from R packages) and the adjusted $p-value$ (column $padj$). Therefore any one can make different list of differential genes based on the selected thresholds. It is common to choose a $log2\ fold\ change\ threshold$ of |1| or |2| and an $adjusted\ p-value$ of 0.01 for instance. Below we are analyzing the distribution of the -value by histogram visualization. Based on the distribution of the p-values (Figure 10) we may notice that:

For the adjusted $p-value$ we have a peak at 0, but we also have a peak close to 1. What do we do in this situation? What is this behavior telling us? Is this something that was proceeded from boxplot and densityplot above?

Some suggestions in this case are:
Figure out why this $p-values$ show this behavior, and solve it appropriately. The $p-value$ close to 1 may indicate increase of gene in response to a drug or they belong to a pathological case! recommendations for this situations are to filter these genes out beforehand (it's not like we are losing any information!). (**Reference:** Ref 1; Ref 2 ; Ref 3)

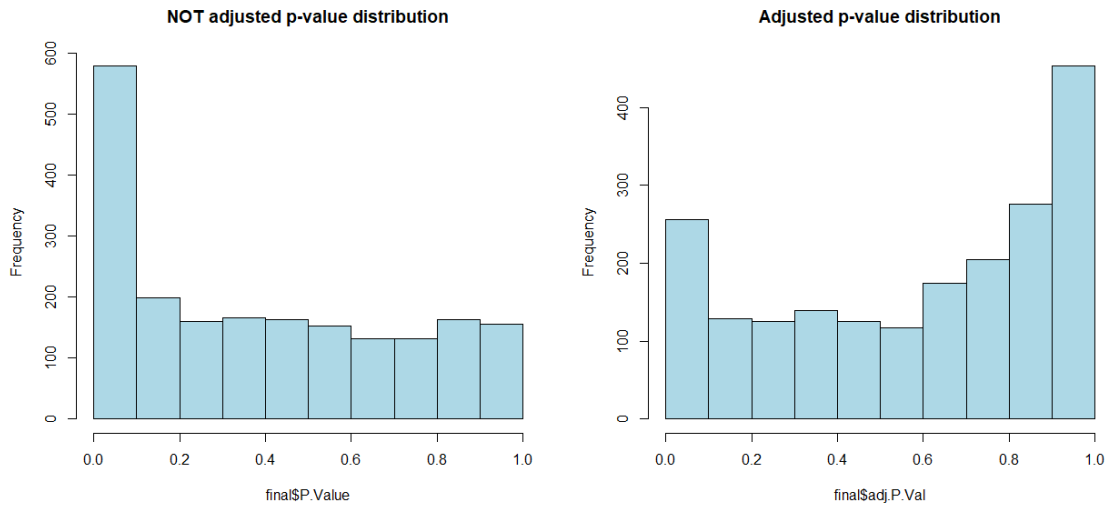


Figure 10: Histogram pf p-value and adj. p-value of all genes for $log2$ transformation

Going through commonly used methods for controlling FWER: $Bonferroni's$ method, $Holm's$ method and multiple testing error measure known as false discovery rate FDR.

We obtain the following results.

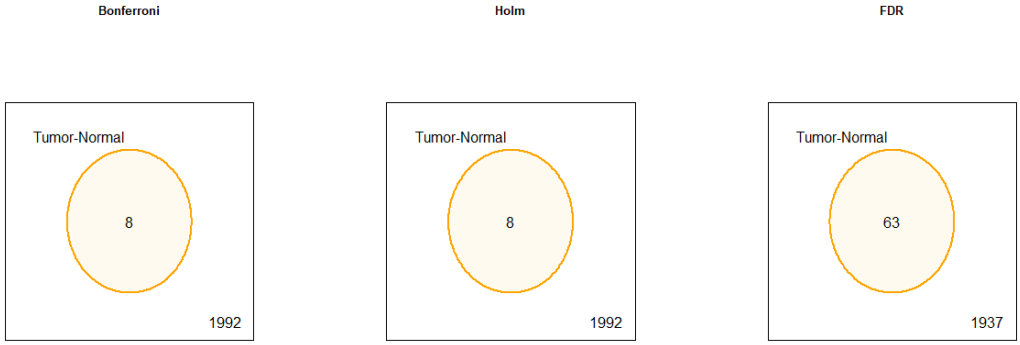


Figure 11: Bonferroni Holm FDR on the original data

Observe above the set of genes when directly applied to the original data which is less than the number of differentially expressed genes obtained after we use $log2$ transformation. We have from FDR: 114 Up and 29 Down and 1857 Not-significant genes.(Figure 12). The ID of these genes are those in rows: Now our subset has 143 genes and 62 samples.

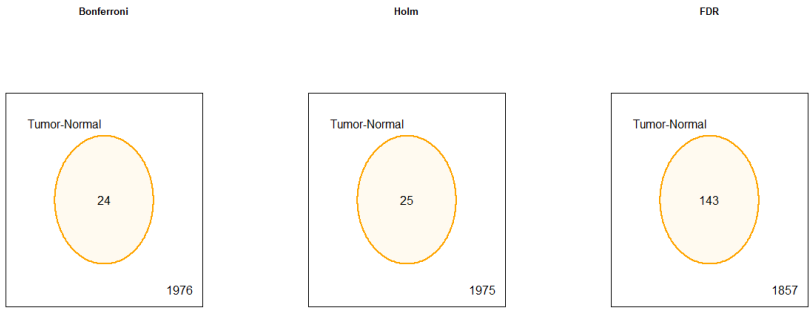


Figure 12: Bonferroni Holm FDR after log2 transformation

So, again here we observe that FDR is less conservative and because the other methods give just 24, 25 genes we will decide to go with FDR decision of considering as deferentially expressed genes 143 from total of 2000. The output from FDR for the log2 transformed data is shown in Figure 13. By transforming the data we went from 3.15% (63 genes) of considered genes to 7.15% (143 genes from which 29 Down and 114 UP). Volcano plot (Figure 13) shows how our genes are represented from each method.

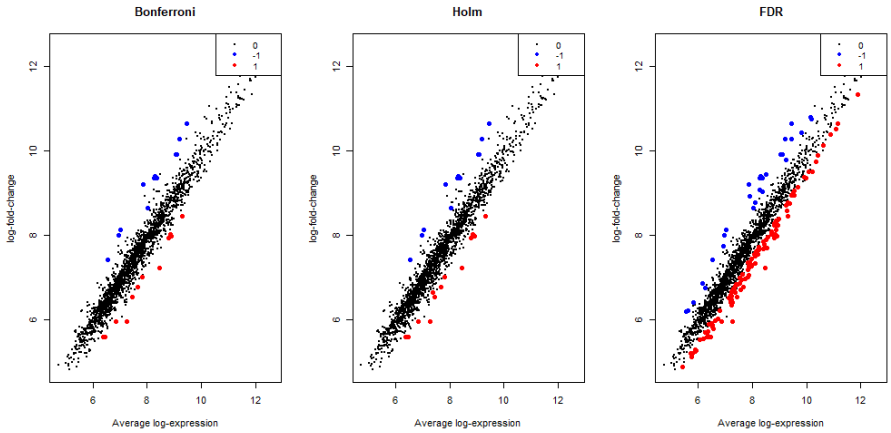


Figure 13: Volcano plot for Bonferroni Holm FDR after log2 transformation

	logFC	AveExpr	t	P.Value	adj.P.Val	B
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
493	-1.533126	8.367340	-6.292594	1.517618e-08	3.035236e-05	9.315900
1671	2.005973	7.260691	5.939360	6.891831e-08	5.586730e-05	7.906059
249	-1.845727	9.451096	-5.893216	8.380096e-08	5.586730e-05	7.723967
1423	-2.081832	7.860515	-5.733383	1.642684e-07	8.213418e-05	7.097333
625	1.452040	8.910696	5.458614	5.139644e-07	2.055858e-04	6.036233
1042	1.392027	7.668984	5.389082	6.835260e-07	2.278420e-04	5.771224

Figure 14: FDR output after log2 transformation

Figure 14 gives the ID of the genes corresponding to differentially expressed genes among Tumor and Normal (sample 1 and sample 2) which were classified by FDR (in total 143 genes).

```
> gene_ID<-which(b3 %in% c(-1,1))
>
> gene_ID
  [1]   26   31   43   47   62   66   72   75  100  102  127  138  141  143  147  187  190
 [18]  201  241  245  249  264  267  281  286  295  343  365  377  391  399  415  444  489
 [35]  493  495  513  515  520  529  549  550  571  581  590  601  619  625  639  648  652
 [52]  698  739  765  780  802  822  830  882  897  964  989  992  994 1002 1042 1047 1060
 [69] 1067 1073 1110 1115 1153 1168 1187 1196 1208 1221 1227 1248 1256 1258 1263 1293 1325
 [86] 1334 1340 1346 1387 1406 1411 1414 1423 1466 1472 1473 1489 1494 1511 1526 1546 1549
[103] 1582 1583 1634 1635 1637 1648 1659 1668 1671 1674 1675 1679 1724 1730 1758 1763 1770
[120] 1771 1772 1785 1808 1836 1839 1843 1867 1870 1884 1886 1892 1897 1900 1904 1920 1935
[137] 1959 1960 1967 1972 1974 1983 1993
> length(gene_ID)
[1] 143
```

Figure 15: Gene ID from FDR output

In R markdown file we have subtracted from the original (2000 genes, remember: log2

transformed dataset) the genes which were classified from FDR as the most differentially expressed genes and created a subset which will be used for further analysis. Figure 14 show a heatmap of the genes before and after we created the subset of the genes selected by FDR.

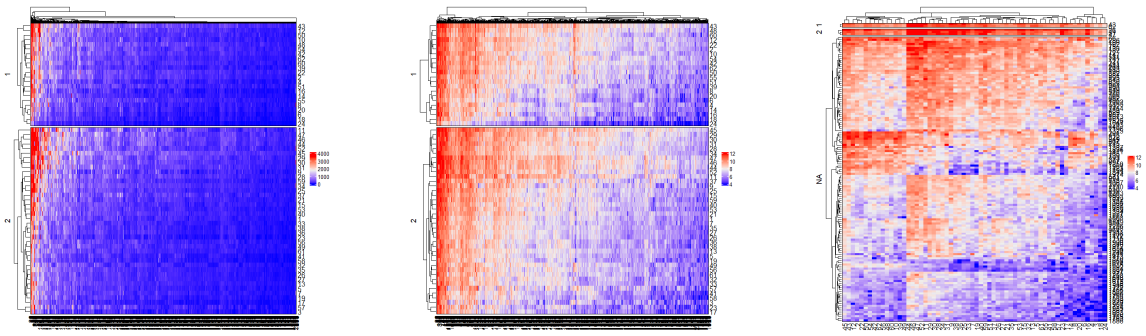


Figure 16: Heatmap of original, log2 and after FDR subset created (from left to right)

2 Question 2

Conduct PCA using the differentially expressed genes identified from (1) and discuss your findings. How many components do you need to explain 80% of the variation in the data?

2.1 PCA with *prcomp*()

To see which genes differ the most across the two groups(normal,tumor), I perform the PCA on the data using the *prcomp*() function

By default, the *prcomp*() function centers the variables to have mean zero. By using the option $scale = TRUE$, we scale the variables to have standard deviation one.

The rotation matrix provides the principal component loadings; each column of **pr.out$rotation** contains the corresponding principal component loading vector.

In this case, the loading can be considered to be the weight of each gene in both the groups.

```
> eig.val <- get_eigenvalue(res.pca)
> eig.val
        eigenvalue variance.percent cumulative.variance.percent
Dim.1  41.610771938     67.114148286                    67.11415
Dim.2   7.983353186     12.876376107                    79.99052
Dim.3   2.197108892      3.543724019                    83.53425
Dim.4   1.485589704      2.396112425                    85.93036
Dim.5   1.250328032      2.016658117                    87.94702
Dim.6   0.907472116      1.463664703                    89.41068
Dim.7   0.625516684      1.008897878                    90.41958
Dim.8   0.533680435      0.860774895                    91.28036
Dim.9   0.426978388      0.688674819                    91.96903
Dim.10  0.372913567      0.601473495                    92.57050
```

Figure 17: PCA table output to log2 transformation data

A screenshot of the variance contribution for each dimension is shown above (Figure 17). We may notice (Figure 17) that the first component explains approximately 67% of the variance and together with component 2 they go up to 80% of variance explained. Associated with the output is the Scree plot (Figure 18) which shows how each Dim is explaining the variation in the data.

The correlation between a variable and a principal component (PC) is used as the coordinates of the variable on the PC. The representation of variables differs from the plot of the observations: The observations are represented by their projections, but the variables are represented by their correlations (Abdi and Williams,2010).

(NOTE: Again here I want to emphasize that all analysis is done with the log2 transformed dataset.) The plot in Figure 19 (left) is also known as variable correlation plots. It shows the relationships between all variables. It can be interpreted as follow: Positively correlated variables are grouped together. Negatively correlated variables are positioned on opposite sides of the plot origin (opposed quadrants). The distance between variables

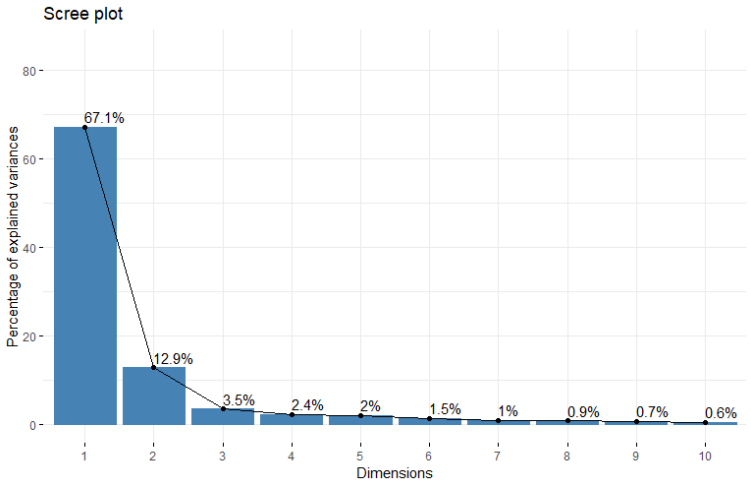


Figure 18: PCA scree plot

and the origin measures the quality of the variables on the factor map. Variables that are away from the origin are well represented on the factor map. The plot to the right shows the individual contributions to dimensions.

A high *cos*2 value indicates a good representation of the variable on the principal component. In this case the variable is positioned close to the circumference of the correlation circle. A low *cos*2 indicates that the variable is not perfectly represented by the PCs. In this case the variable is close to the center of the circle. It's possible to color variables by their cos2 values using the argument col.var = "cos2". This produces a gradient colors. In this case, the argument *gradient.cols* can be used to provide a custom color. For instance, gradient.cols = c("white", "blue", "red") means that: variables with low cos2 values will be colored in "white" variables with mid cos2 values will be colored in "blue" variables with high cos2 values will be colored in red. We may observe a clear division in two groups for the individuals (genes). From Figure 2 when clustering in 2 clusters we observe a clear division of genes but when increasing to 3 clusters the clusters are not clearly divided (more graphs are in R Markdown file. See Appendix).

2.2 Variable contribution

Variables that are correlated with PC1 (i.e., Dim.1) and PC2 (i.e., Dim.2) are the most important in explaining the variability in the data set. Variables that do not correlated with any PC or correlated with the last dimensions are variables with low contribution and might be removed to simplify the overall analysis. The larger the value of the contribution,

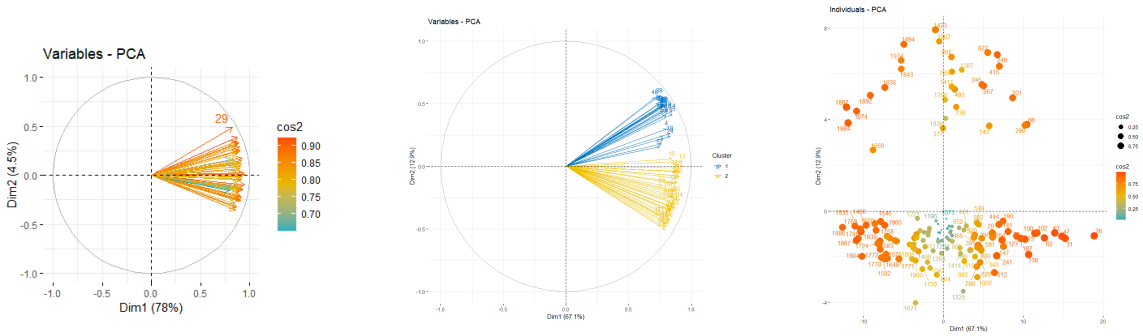


Figure 19: PCA variable and individual contribution to dimensions

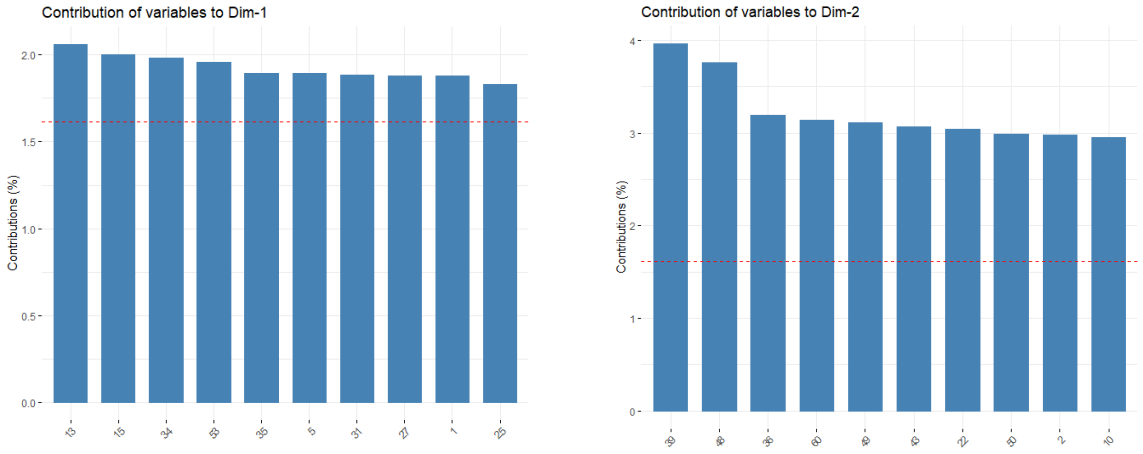


Figure 20: Top 10 contribution to Dim1 and Dim2

the more the variable (sample) contributes to the component.

Link between the variable and the continuous variables (R-square)

	correlation	p.value
13	0.9263233	1.200862e-61
15	0.9122200	1.673564e-56
34	0.9085283	2.676463e-55
53	0.9029555	1.417047e-53
35	0.8880039	2.015341e-49
5	0.8878490	2.209094e-49
31	0.8857127	7.732266e-49
27	0.8848445	1.277469e-48
1	0.8845917	1.477437e-48
5	0.8727034	9.624798e-46
41	0.8719548	1.415931e-45
32	0.8669709	1.741780e-44
33	0.8652106	4.125369e-44

Figure 21: Link between the variable and the continuous variables (R-square)

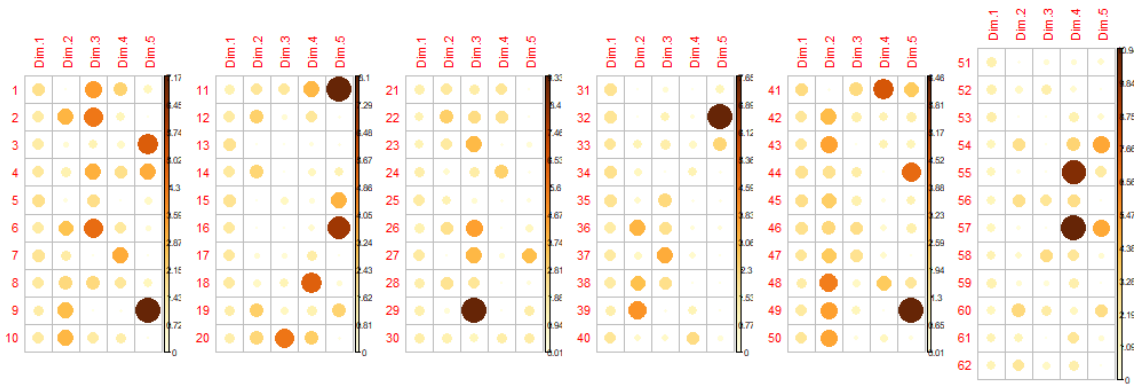


Figure 22: PCA variable contribution and first 5 dimensions

As we may observe from the contribution plots (Figure 20) there are different samples which contribute to Dim1 and Dim 2. What is clearly observed is the fact that for Dim 1 and Dim2 the number of genes contributing to is larger compared to other dimensions. This contributions may be also seen by a barplot of contributions only to Dim 1 , only to Dim 2 and Dim1 and Dim2 together (For more see Appendix) To help understanding the above graph below is shown the bar graph for the contribution of samples to each of the two dimensions (Dim1 and Dim2).

Figure 22 shows the contribution of samples (variables) to each of the first 5 dimensions

of PCA. This result will be used in Question 4 to create a subset of the data and then use it further to increase accuracy in cluster analysis and prediction. The color and the size of the circle shows the contribution of the variable (row) to the dimension (column). The dark color shows a high contribution and the light color shows a lower contribution of the variable to the dimension.

Note also that, the function dimdesc() [in FactoMineR], for dimension description, can be used to identify the most significantly associated variables with a given principal component.

So, the first 2 dimensions will explain up to 80% **of the variance of the data.**

3 Question 3

Use an appropriate clustering techniques, cluster the tissue samples using all differentially expressed genes. Discuss why the clustering approach is appropriate and list two advantages and limitations of your chosen approach. How were the number of clusters selected? Report the classification accuracy of your chosen approach.

3.1 Cluster Analysis

The distance matrix is a key element when working with clusters since it helps to identify the observations which are close to each other and group them into one cluster. In the previous parts we have used the *heatmap* which is also based in one cluster methodology which is the hierarchical cluster. The distance matrix show some clear patterns of genes which are close and it also shows a division of genes/samples starting from 2 up to 4 clusters.

At this part we have considered two main cluster techniques: $Hierarchical$ and $kmeans$.

3.1.1 Hierarchical clustering

If we don't know how to reason and decide on the number of clusters we may use an alternative methodology which is Hierarchical Clustering . Hierarchical Clustering has an added advantage that it produces a tree based representation of the observations, called a Dendogram. To choose the number of clusters we just draw horizontal lines across the dendogram. We can form any number of clusters depending on where we draw the break point.

Implementing hierarchical clustering involves one obvious issue. How do we define the dissimilarity, or linkage. Some of the options are:

$CompleteLinkage$: Largest distance between observations
$SingleLinkage$: Smallest distance between observations
$AverageLinkage$: Average distance between observations
$Centroid$: distance between centroids of the observations.We may create a dendogram and then also visualize the clusters .

Hierarchical clustering can be divided into two main types: agglomerative and divisive.

Agglomerative clustering: It's also known as AGNES (Agglomerative Nesting). It works in a bottom-up way. That is, each object/observation is initially considered as a single element cluster (leaf). At each step of the algorithm, the two clusters which are the most similar are combined into a new bigger cluster (nodes). This procedure is iterated until all points are member of just one single big cluster (root). The result is a tree which can be plotted as a dendrogram.

Divisive hierarchical clustering: It's also known as DIANA (Divise Analysis) and it

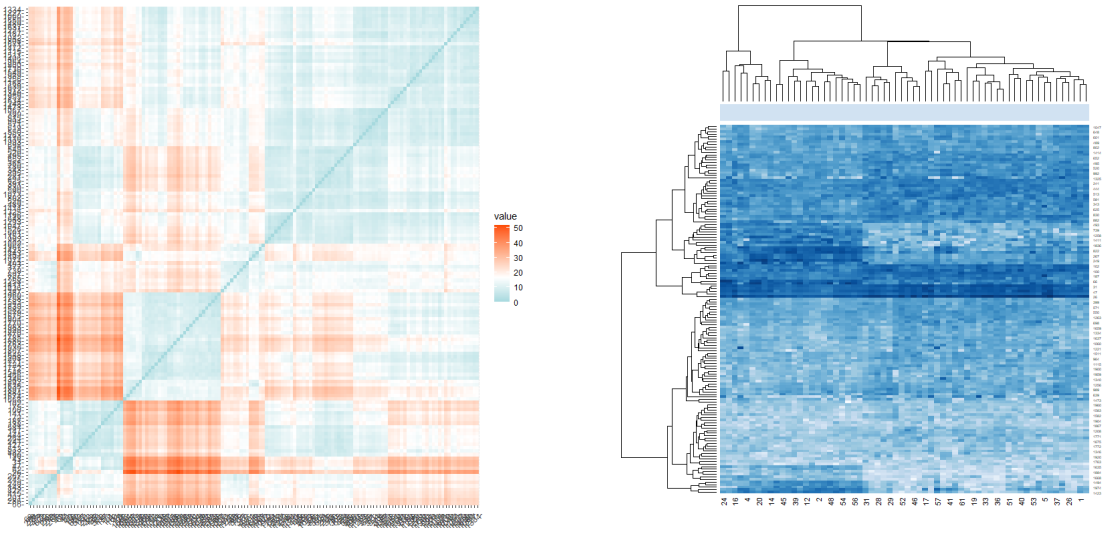


Figure 23: Distance matrix and clusters (log2 transformed data)

works in a top-down approach. The algorithm is an inverse order of AGNES. It begins with the root, in which all objects/observations are included in a single cluster. At each step of iteration, the most heterogeneous cluster is divided into two. The process is iterated until all objects are in their own cluster (see figure below).

These two functions behave very similarly; however, with the *agnes* function you can also get the *agglomerative coefficient*, which measures the amount of clustering structure found (**values closer to 1 suggest strong clustering structure)**.

As we can see from Figure 25 the *ac* value obtained from agnes is 0.76 which is a value close to 1 and also *dc* value obtained from diana is very close to 1 (0.88) thus we can say the clusters are accurately formed and are valid. And also he visualizations suggest : Agnes has 2 to 4 clusters where the differences are observed. Diana has 2 clear clusters and up to 6 very small clusters. No matter which approaches to take either DIANA or AGNES both will give you clusters with same meaning just the number of entries in the clusters might differ.

As from the above results a number of cluster from 2 up to 4 will give accurate results in gene classification for the dataset considered. Further we will also observe suggestions from the kmeans methodology and advice how to proceed.

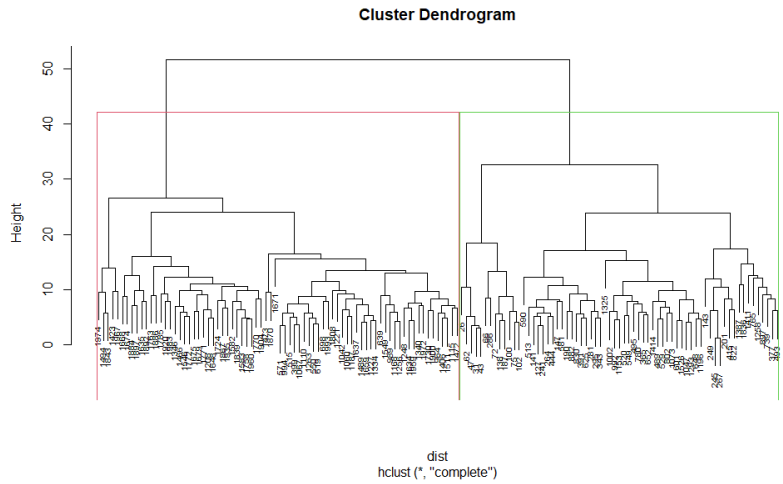


Figure 24: Hierarchical clustering dendogram with 2 clusters

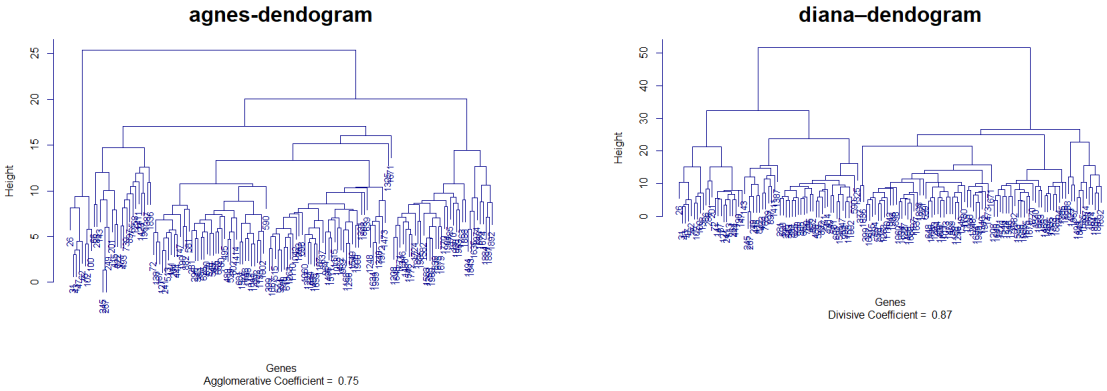


Figure 25: Distance matrix and clusters (log2 transformed data)

3.1.2 Kmeans

Determining Optimal Clusters is one of the challenges when using *kmeans*. To help taking a decision we may use three most popular methods for determining the optimal clusters, which includes:
Elbow method
Silhouette method
Gap statistic
From a start visualization we may observe that our genes are clearly organized in different

clusters. And using a number of clusters from 2 up to 5 the clusters remain clear and we do not have any cross location (overlapping) of the clusters. (Figure 26) But for a better decision we have also considered the below methods to decide on the optimal number of clusters.

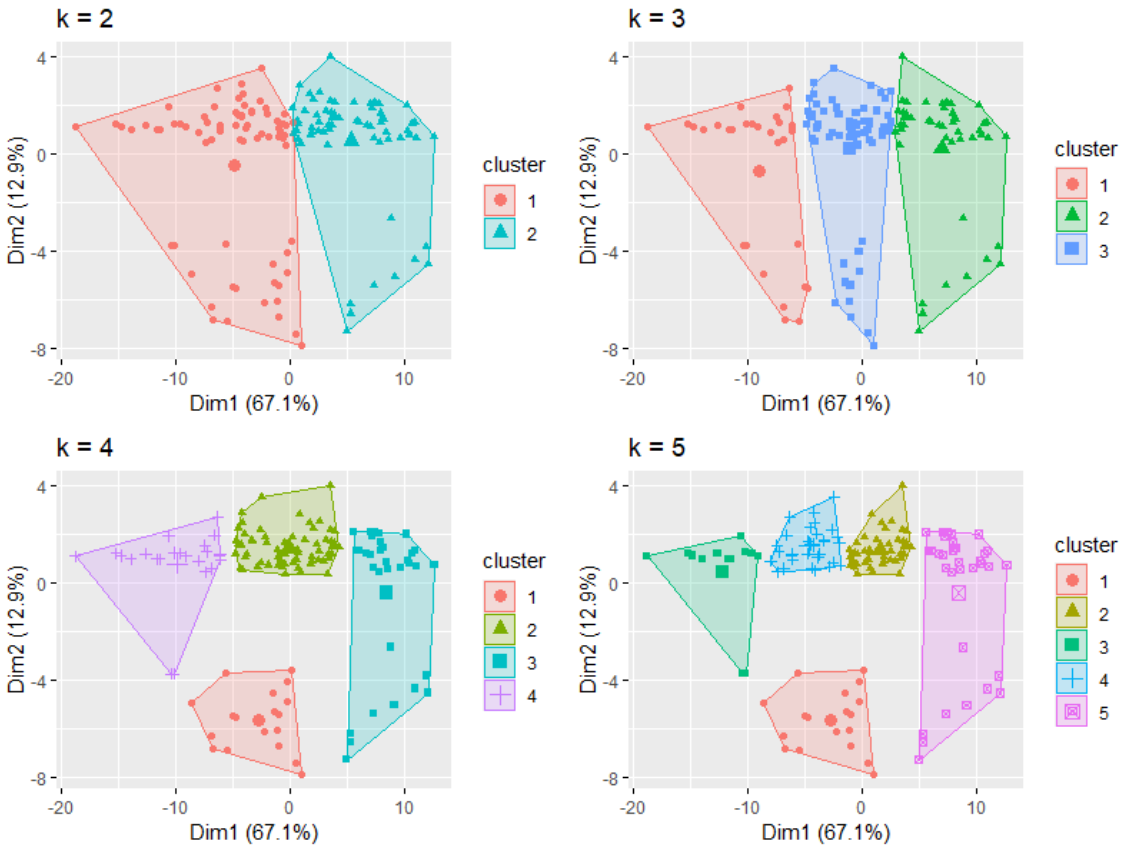


Figure 26: Kmeans for 2:5 clusters

Based on Figure 27 results for the optimal number of clusters we may advice: The WSS suggest using 2 or 3 clusters, the *silhouette* suggest 2 clusters and the *Gapstatistic* method suggest 1 or 2 cluster. Analysing also the *kmeans* vizualizations above we may suggest at this step that 2 clusters will perform good in classification of the genes.

End note cluster: Clustering is an unsupervised machine learning algorithm in which we compute analytics mostly without a pre-defined aim to understand the relationships between the data. Once we get the understanding and trends in the data we can accordingly take necessary actions and data-driven decisions. Both methodologies suggested here have

their advantages and disadvantages. So, a step by step reasoning and further investigation of the data will help on deciding the optimal number of clusters used for a "best" classification of observations.

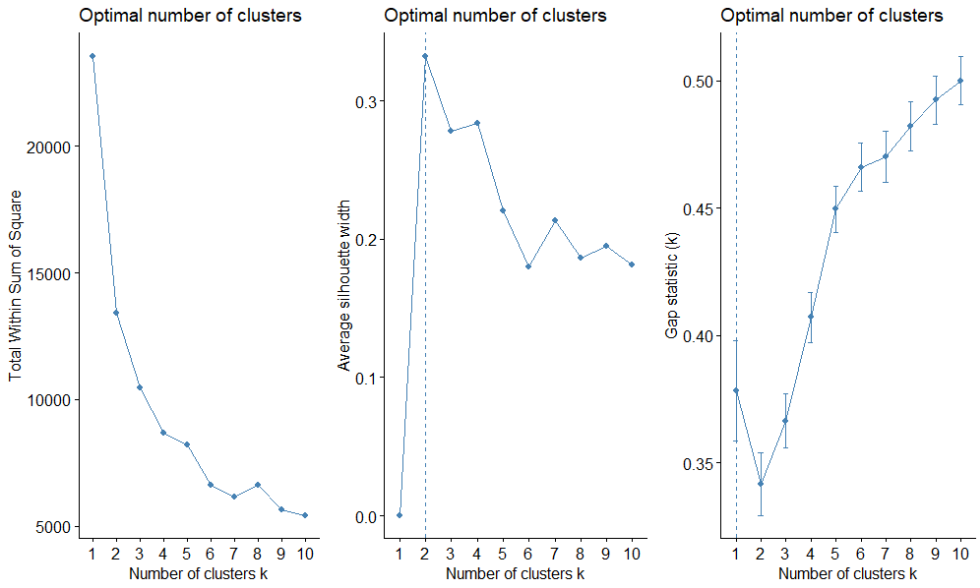


Figure 27: Optimal number of clusters from FDR subset

4 Question 4

Can you increase the clustering accuracy through dimension reduction (either built-in in the clustering framework or conducting it prior to cluster analysis), variable selection (e.g., top few hits), or a combination of both? Justify your answer!

4.1 Different approach

The high amount of data in genomic dataset is extremely important especially in the pre-processing phase when we want to extract as much information as we can. But the high dimension of information are not always good. It happens that even when using machine learning methods at a certain point, increasing the number of features or dimensions can decrease the model accuracy.

Depending on the problem and the nature of the data it is possible to increase the clustering accuracy through dimension reduction or variable selection. Some times an ensemble approach may help to achieve a better clustering performance.

4.1.1 First approach -PCA

Dimensionality reduction is defined as way to reduce the complexity of a model and avoid over-fitting. In our case we may use a feature selection approach which is done by selecting a subset of the original features (in our case samples). Based on the above results of PCA we saw that it suffices two components to explain more than 80% of variation in data. So, we may select a subset of the features (samples) (observe correlation plot of Dim1 to Dim 5 and 62 samples). By selecting those samples which contribute mostly to Dim1 and Dim 2 we may then start analyzing again the clustering approaches. Let's bring at our attention Figure 22. In this visualization we may select those samples who are contributing more to the first and second dimmensions (which together explain more than 80% of the variation). Returning to PCA results from $factoextra$ and $FactoMineR$ packages we may obtain the correlations of variable and dimensions (Use: modelvarcor) or contribution of the variables (Use: modelvarcontrib) (Reference: PCA packages in R)

Bring to your attention Figure 21, which shows the results of correlation and link between variables and dimensions. Based on these values we may proceed and construct a subset based on a given threshold of the correlation. In this case I have considered correlation coefficient greater than 0.85. And the sample ID where this correlation is $>= 0.85$ is compound of 18 samples. Which are almost 30% of the total number of samples (62

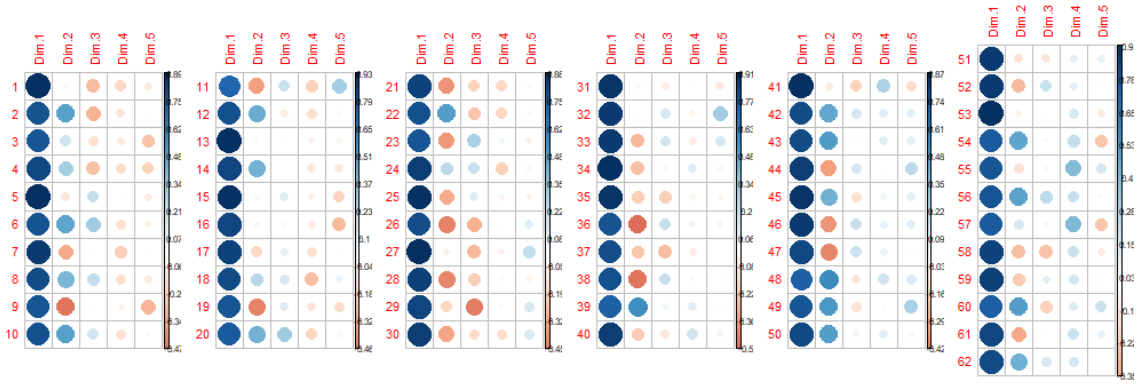


Figure 28: Variable correlation from PCA (first 5 Dim)

samples). So, with 30% of the samples we aim to achieve a higher accuracy for clustering genes in our dataset.

4.1.2 Second approach- Random Forest (RF)

Another approach may be the use of a random forest algorithm which will analyse the feature importance (samples). It will suggest which variables (samples) are of high importance and will suggest to create further algorithms for analysis and prediction purpose.

5 Question 5

Implement a dimension reduction technique, variable selection, or a combination of both on the deferentially expressed genes and report your clustering performance. Make sure you describe your analysis steps you used and what it does.

5.1 Scheme of dimension reduction using PCA

Following the reasoning from Question 4 I decided to reduce the number of samples used based on the correlation to Dim 1 as the one which has the highest impact on explaining the variance. The scheme followed is explained in Figure 29.

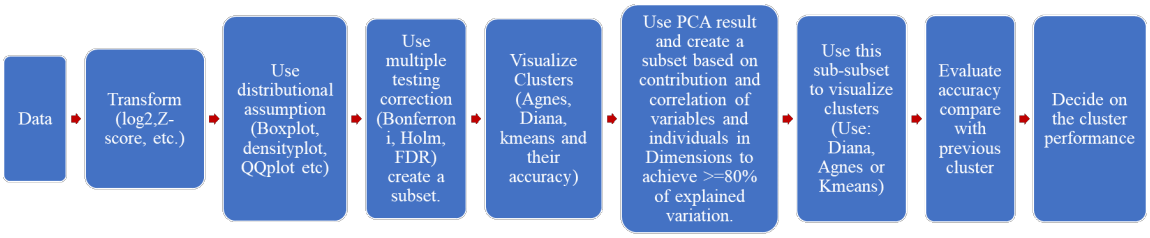


Figure 29: Dimension Reduction scheme

All the code in R to extract from the set (log2 transformed: 114x62) is found in R markdown file. Below is a simple way how to extract subset from the subset(log2 and FDR) of original data.

```
which(res.pca$var$cor[,1]>=0.85)
# which samples have correlation in PCA greater or equal to 0.85 for Dimenssion 1
# 1  5  7 13 15 17 25 27 31 32 33 34 35 40 41 51 52 53
# is the % of samples used after reduction.
# Almost 30% of samples is used and the performance is higher.
Colon_subset<-log2_col.matrix[gene_ID,]
Dim_red<-Colon_subset[,samp_ID] # subset of features (samples) from 62 now we have 18
head(Dim_red)
```

5.2 Basic test for normality assumption

Below are the graphical output for the normality assumption in our reduced subset.

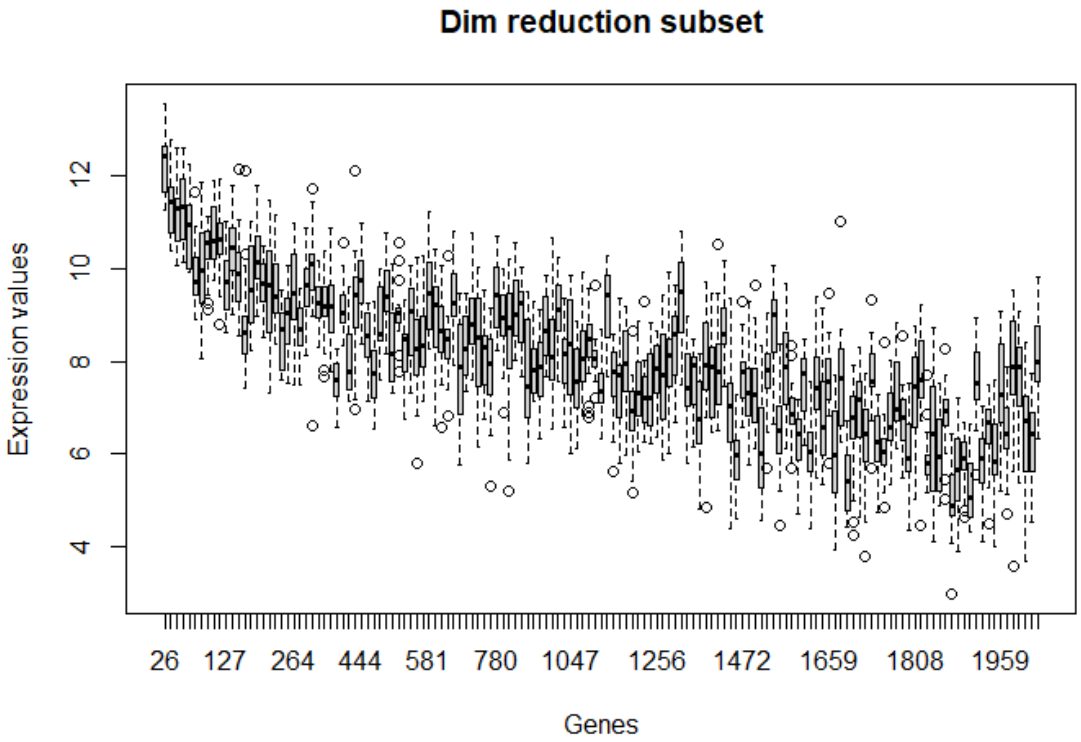


Figure 30: Boxplot after dimension reduction

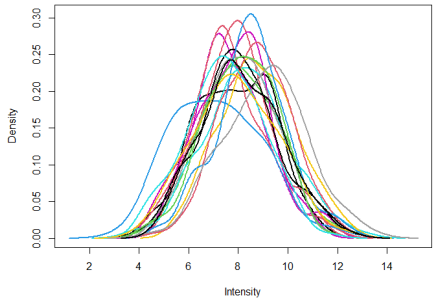


Figure 31: Density plot after dimension reduction

5.3 Clustering after dimension reduction

5.4 Hierarchical and Kmeans Clusters

The results after dimension reduction for diana and agnes approach are shown in Figure 32. The coefficients respectively for diana and agnes are increased (from 0.75 to 0.79 agnes and from 0.87 to 0.91 for diana). Observing the clustering process for *kmeans* were PCA is also visualized we also have an improvement of Dim 1 from 67% to 80.6%. Visually comparing our first try of clustering using *kmeans* (Figure 26) and after dimensional reduction (Figure 33) we do not have a significant change in the performance, since our individuals (genes) were correctly classified to a cluster in both steps.

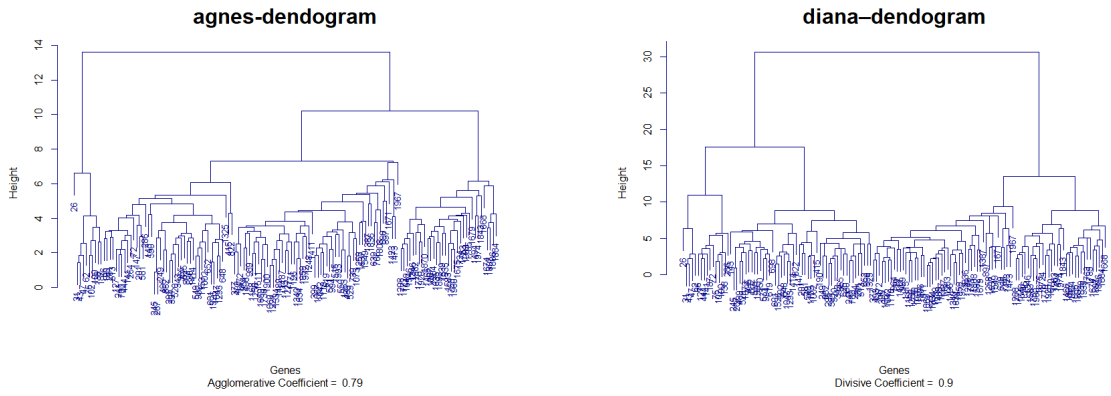


Figure 32: Diana and Agnes after dimension reduction)

Dimension reduction is an important step in cluster analysis. Apart making the high dimensional data addressable it also reduces the computational cost, and can also provide users with a clearer visualization of the data of interest. Below is the final visualization of the heatmap and clusters for both genes and samples. These results may be used further as a training set for all the data and classify each gene in the appropriate cluster.

Even after dimension reduction we notice two clusters of individuals (genes) which contributes to the explanation of the variance (Figure 36). Size of the circles shows the contribution value and the color shows the value of cos2. To the left and right we observe the creation of two clouds.

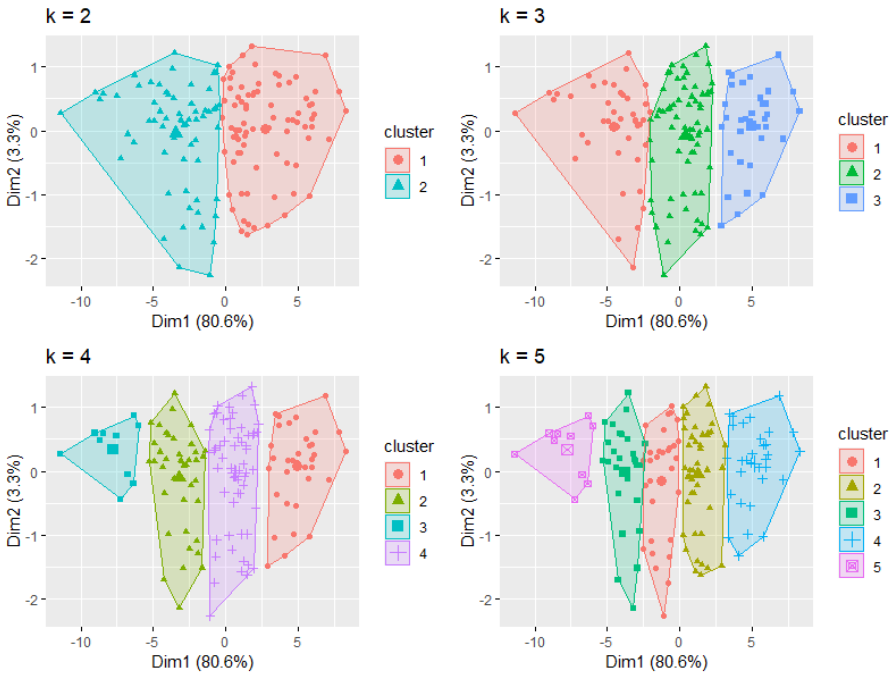


Figure 33: Kmeans after dimension reduction

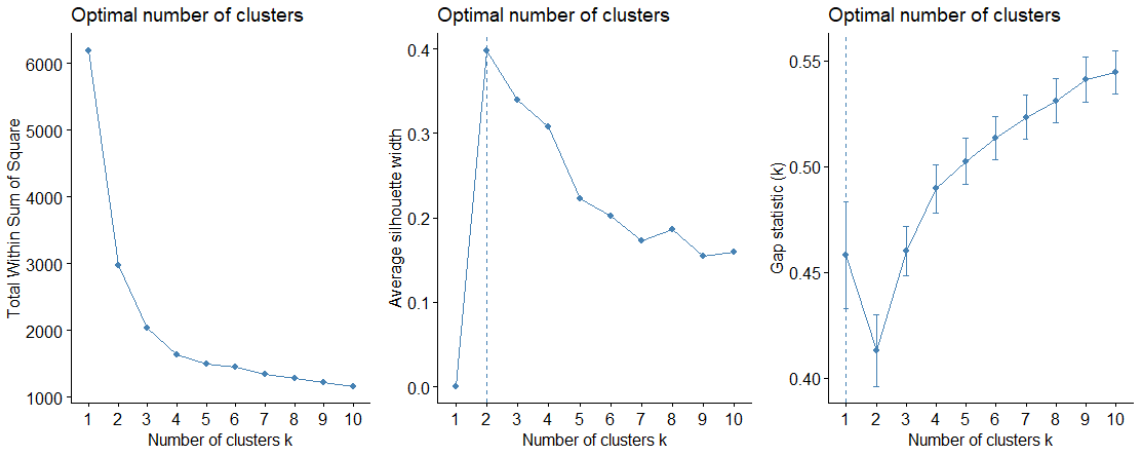


Figure 34: Optimal number of cluster after dimension reduction

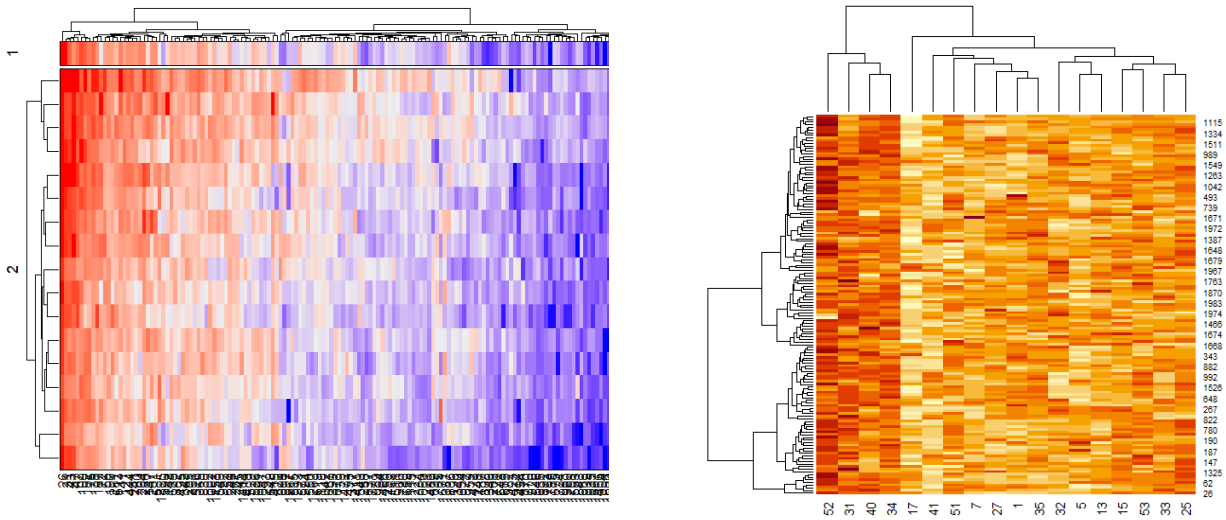


Figure 35: Heatmap after dimension reduction

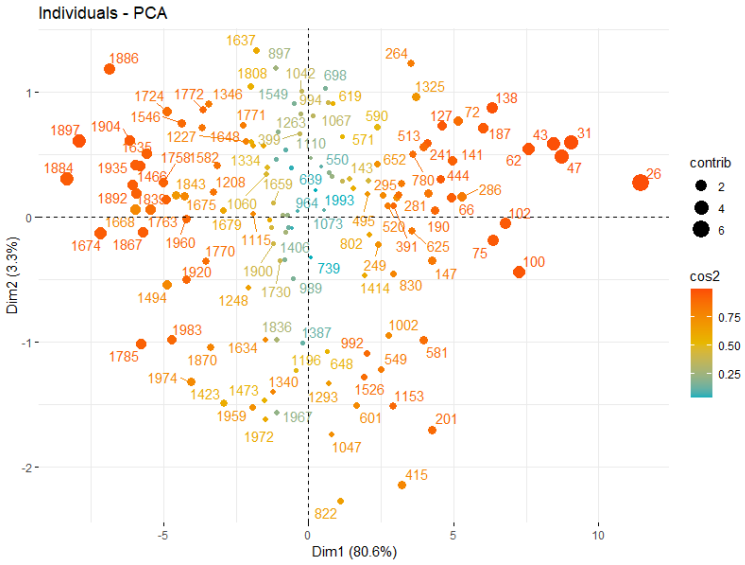


Figure 36: Individuals contribution after dimension reduction

6 Appendix

R codes with the analysis included in this material and more may be downloaded from here: R Markdown Code Output-Github