In [1]:
```
head(mtcars)
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225	105	2.76	3.460	20.22	1	0	3	1

In [2]:
```
heatmap(mtcars)
```

```
Error in heatmap(mtcars): 'x' must be a numeric matrix
Traceback:

1. heatmap(mtcars)
2. stop("'x' must be a numeric matrix")
```

It is throwing an error that the data should be in numeric matrix form

In [3]:
```
class(mtcars)
```

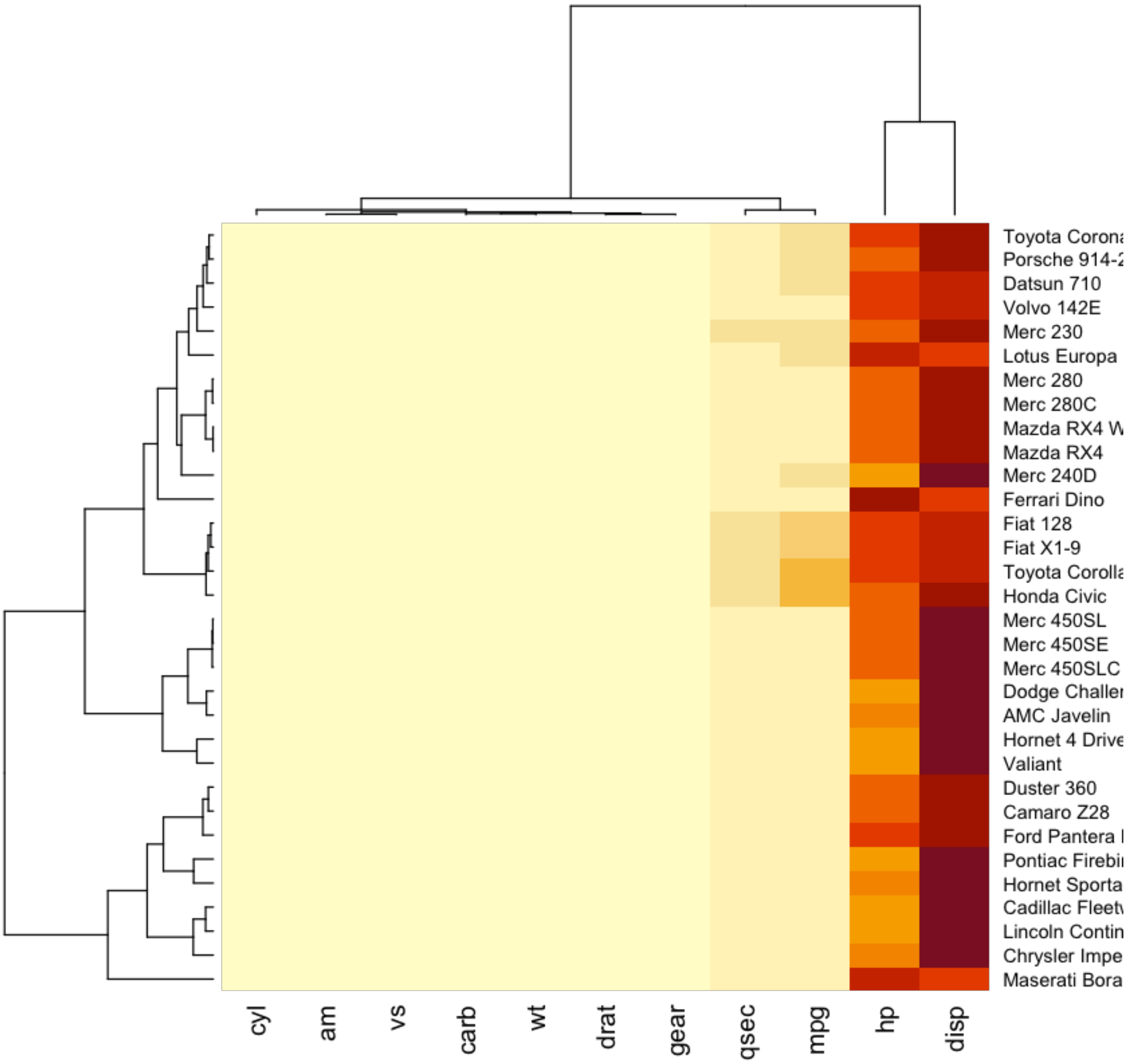
'data.frame'

To convert the data frame into numeric matrix

In [4]:
```
mtcars_mat=data.matrix(mtcars)
```

Now we are going to visualize the data

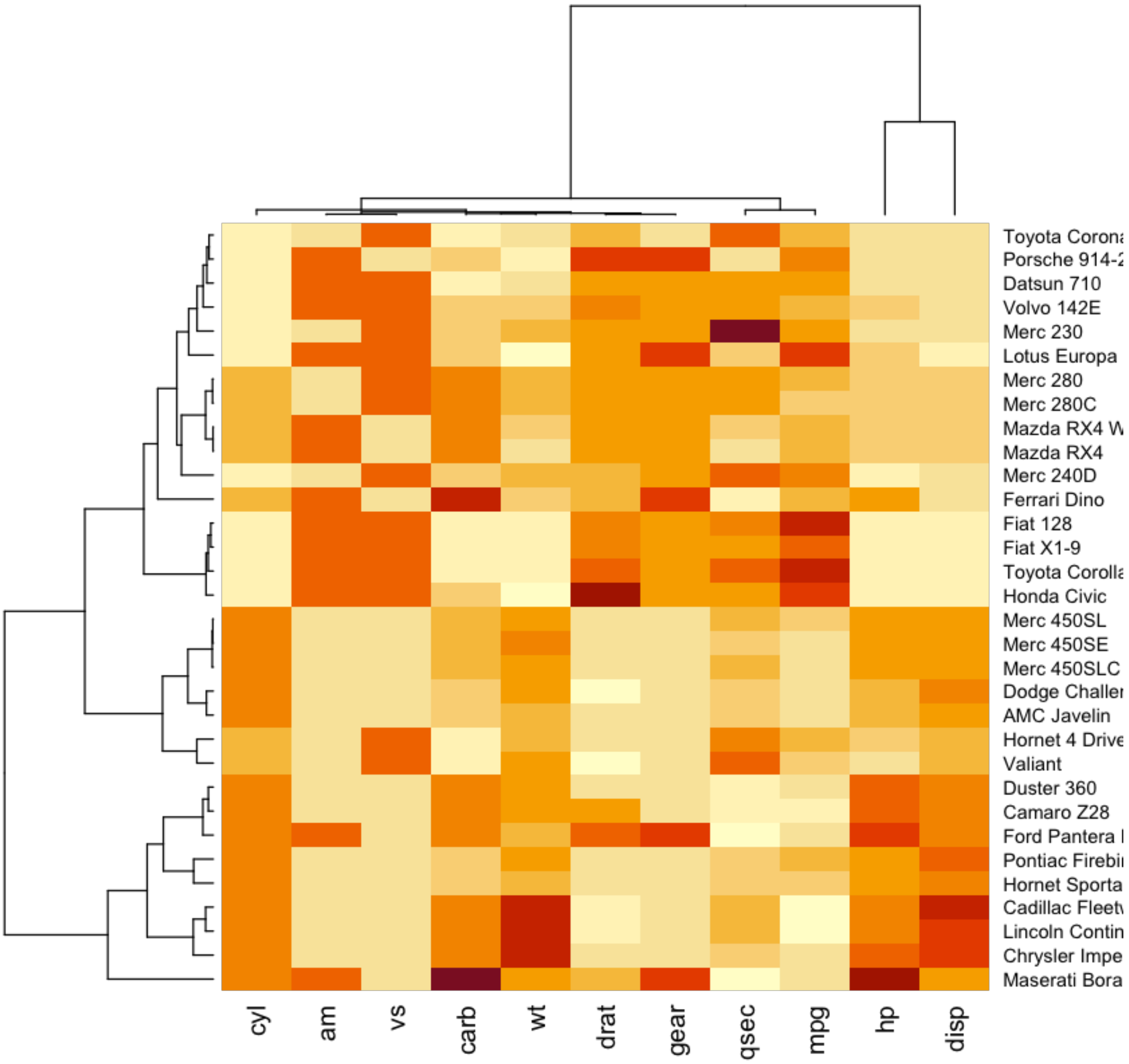
In [6]:
```
heatmap(mtcars_mat)
```



Here as the data values are irregular, so we are scaling the data to get clear visualization

In [7]:

```
heatmap(mtcars_mat, scale="column")
```



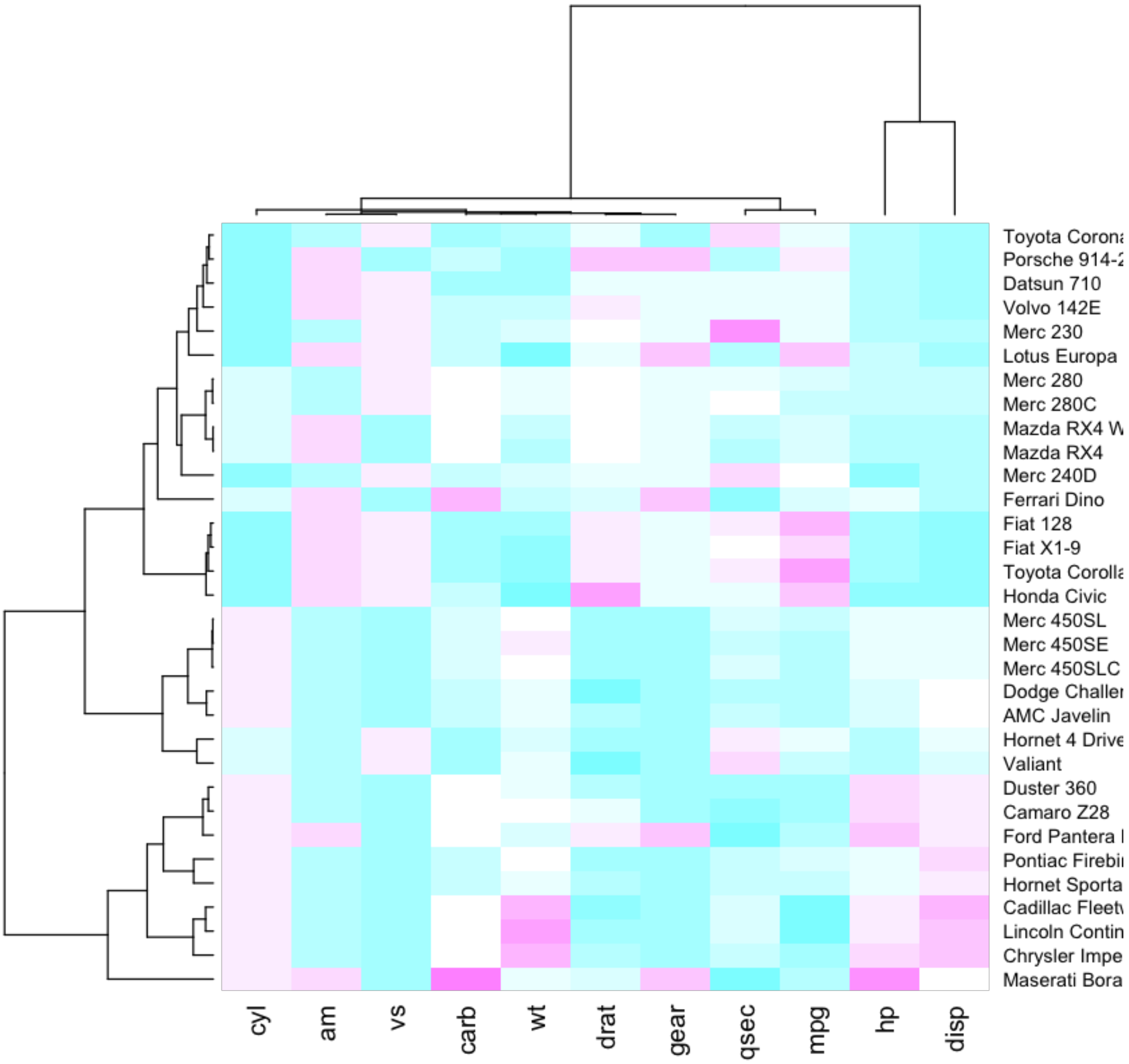
Light yellow color represents low values

Red color represents high values

Dendograms are used to compare the different variables

To visualize with more colors

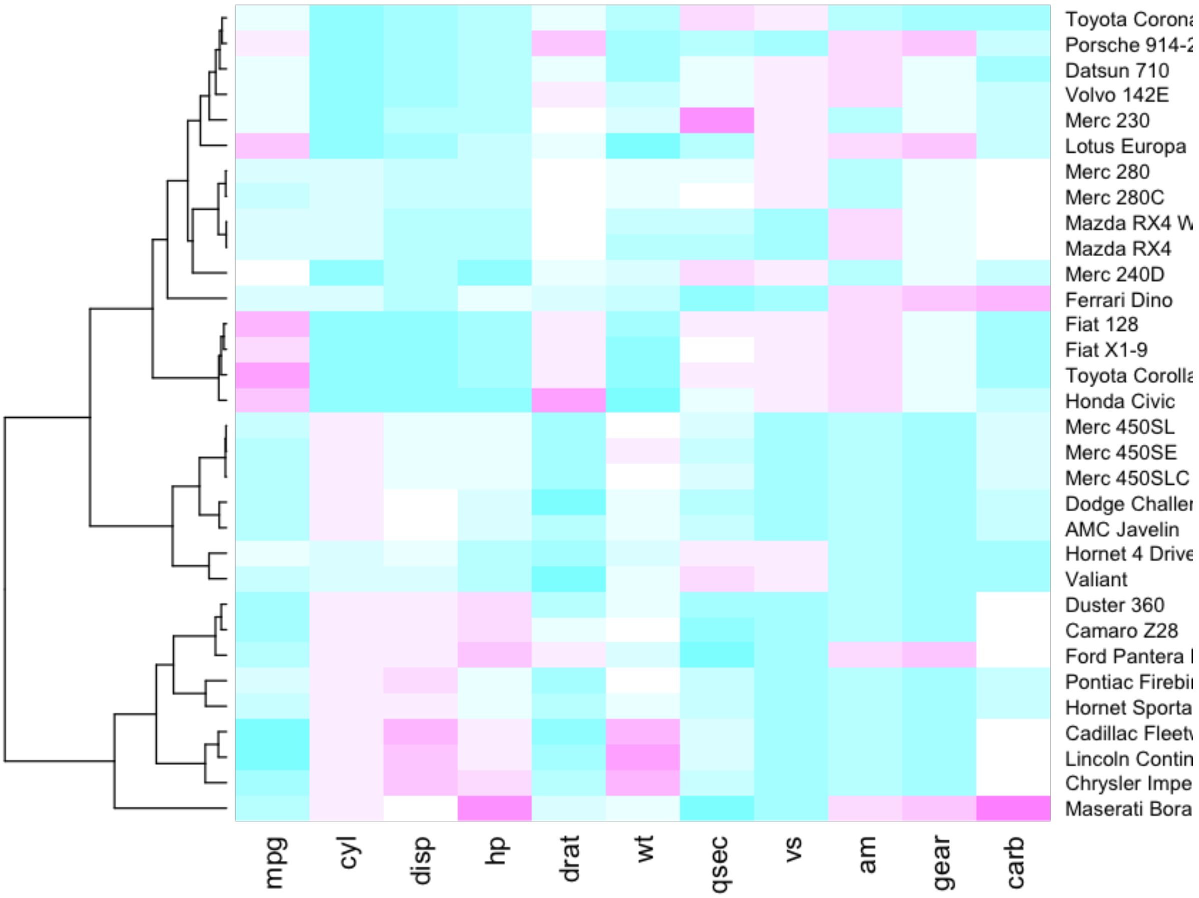
```
In [8]: heatmap(mtcars_mat, scale="column", col = cm.colors(n=15))
```



Rowv and Colv can be set to NA to remove dendrograms and the data won’t be reorganised according to the clustering method.

Remove dendograms as this doesn't make sense for this dataset

In [9]:
```
heatmap(mtcars_mat,
        scale = "column",
        col = cm.colors(15),
        Colv = NA)
```



In [10]:
```
colnames(mtcars_mat)
```

```
'mpg'  'cyl'  'disp'  'hp'  'drat'  'wt'  'qsec'  'vs'  'am'  'gear'  'carb'
```

Clean up the environment

In [11]:
```
rm(list = ls())
```

In [12]:

```
install.packages("gplots")
```

```
Updating HTML index of packages in '.Library'
Making 'packages.html' ... done
```

In [13]:

```
library(gplots)
```

```

Attaching package: ‘gplots’

The following object is masked from ‘package:stats’:

    lowess

```

In [14]:

```
?heatmap.2
```

In [15]:

```
rawdata <- read.csv("https://raw.githubusercontent.com/ab604/heatma
```

In [16]:

```
head(rawdata)
```

Row_ID	X1.Control	X2.Control	X3.Control	X1.Treatment	X2.Treatment	X3.Treatment	
1	7.152522	7.163227	7.112041	6.912676	6.882109	6.918670	0.00
2	6.639397	6.614227	6.594811	6.365425	6.349997	6.411068	0.00
3	3.678131	3.776133	3.813182	2.398460	2.088798	2.172404	0.00
4	5.040845	5.007151	4.686440	3.425154	3.516301	3.657402	0.00
5	6.915152	6.810610	6.897310	7.124722	7.211372	7.269469	0.00
6	4.570305	4.658906	4.819562	5.295718	5.119475	5.134975	0.00

cleanup data

In [19]:

```
rawdata=rawdata[,2:7]
```

In [20]:

```
colnames(rawdata)
```

'X1.Control' 'X2.Control' 'X3.Control' 'X1.Treatment' 'X2.Treatment'
'X3.Treatment'

In [50]:
```
head(rawdata)
```

Control_1	Control_2	Control_3	Treatment_1	Treatment_2	Treatment_3
7.152522	7.163227	7.112041	6.912676	6.882109	6.918670
6.639397	6.614227	6.594811	6.365425	6.349997	6.411068
3.678131	3.776133	3.813182	2.398460	2.088798	2.172404
5.040845	5.007151	4.686440	3.425154	3.516301	3.657402
6.915152	6.810610	6.897310	7.124722	7.211372	7.269469
4.570305	4.658906	4.819562	5.295718	5.119475	5.134975

In [51]:
```
str(rawdata)
```

```
'data.frame':	63 obs. of  6 variables:
 $ Control_1  : num  7.15 6.64 3.68 5.04 6.92 ...
 $ Control_2  : num  7.16 6.61 3.78 5.01 6.81 ...
 $ Control_3  : num  7.11 6.59 3.81 4.69 6.9 ...
 $ Treatment_1: num  6.91 6.37 2.4 3.43 7.12 ...
 $ Treatment_2: num  6.88 6.35 2.09 3.52 7.21 ...
 $ Treatment_3: num  6.92 6.41 2.17 3.66 7.27 ...
```

changing the column names to control and treatment and removing id column

In [21]:
```
colnames(rawdata)=c(paste("Control", 1:3, sep="_"),
                   paste("Treatment",1:3, sep="_"))
```

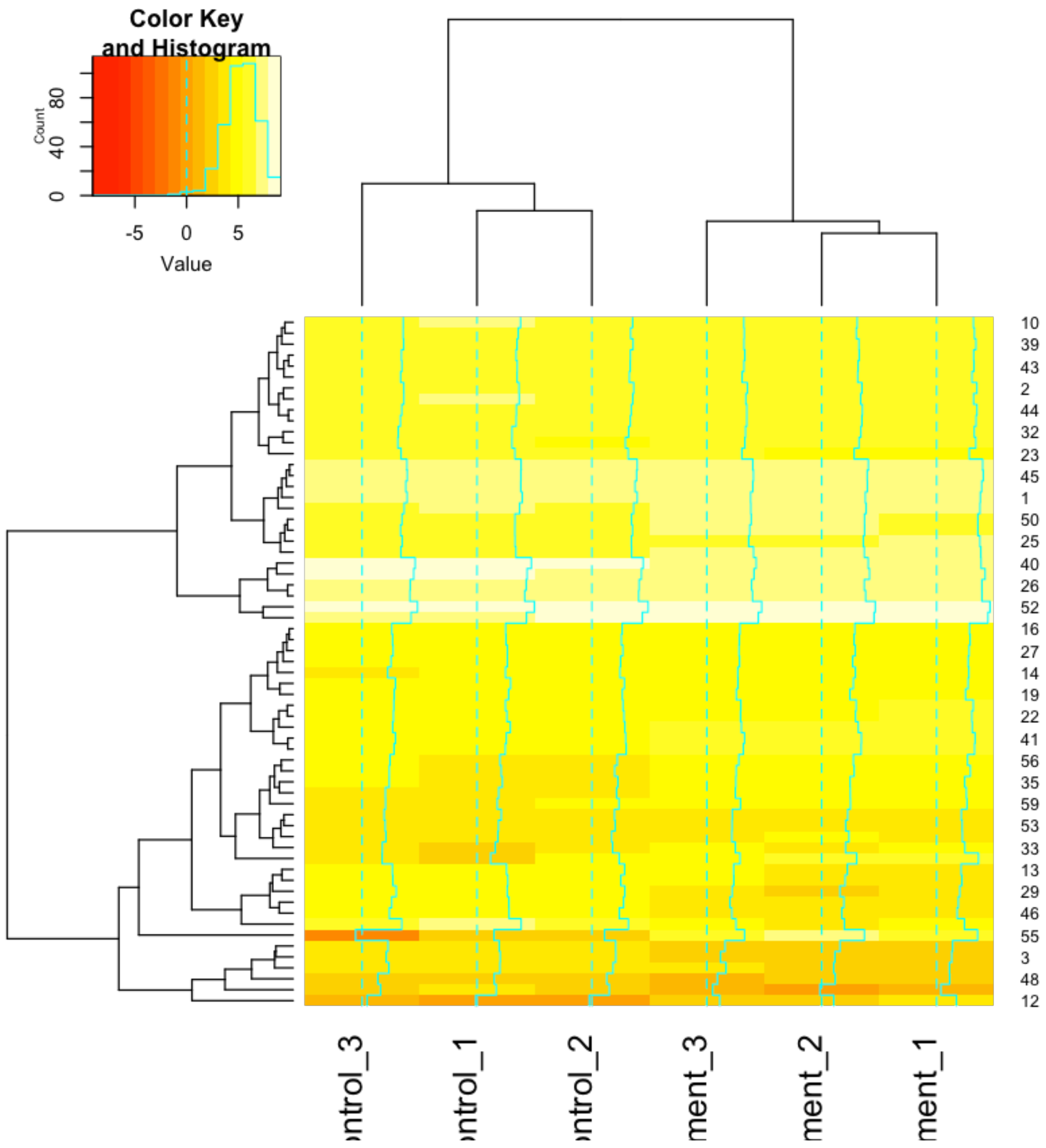
convert to matrix

In [22]:
```
data_mat=data.matrix(rawdata)
```

create a heat map.2

using the basefunction heatmap.2 to visualize data

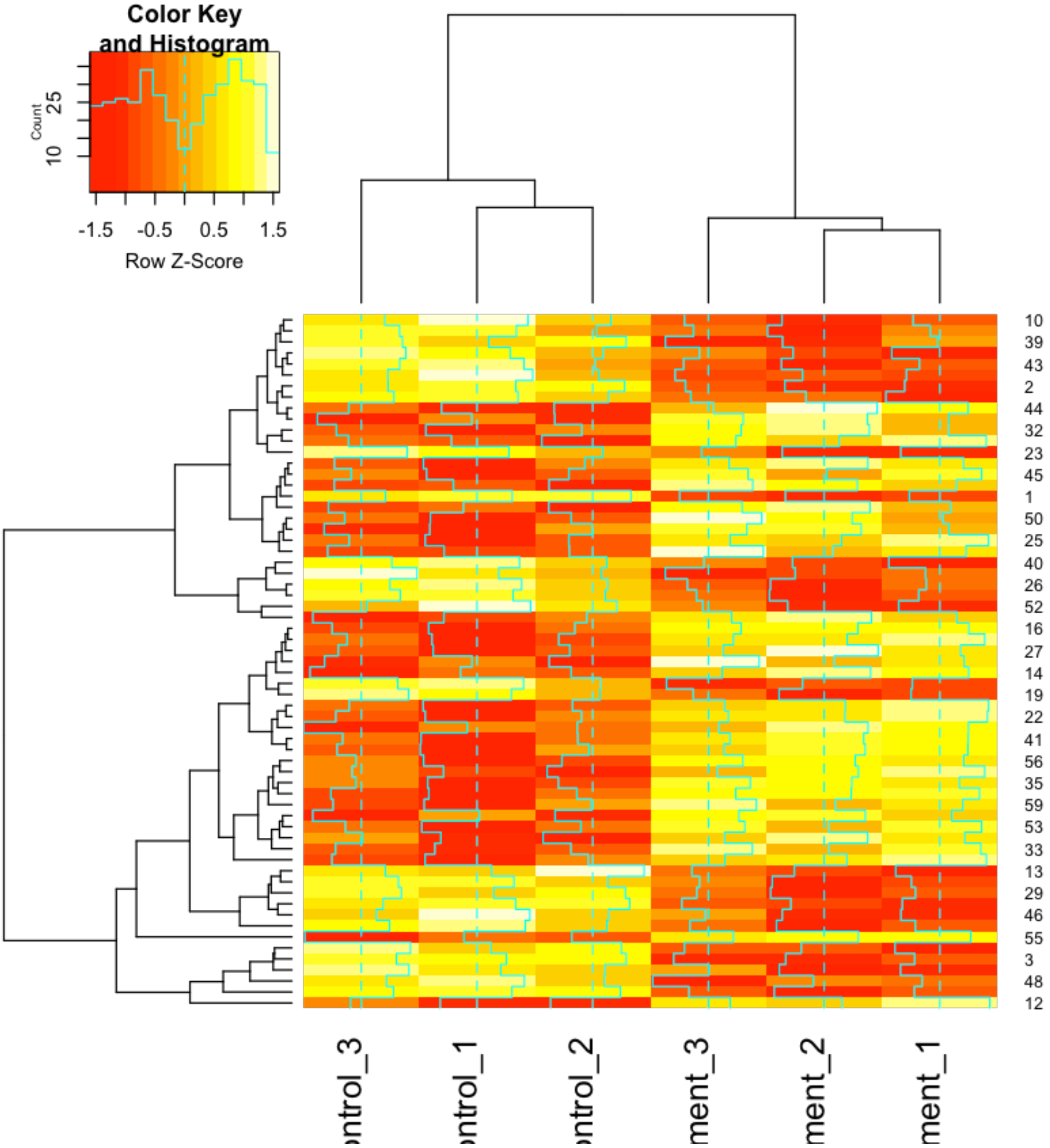
```
In [46]: heatmap.2(data_mat)
```



Here lighter color are representing high values

In [24]:

```
heatmap.2(data_mat, scale = "row")
```



Scale the data before visualising and also to see how the protein responded to a particular treatment

In [25]:

```
data_scaled=t(scale(t(data_mat)))
```

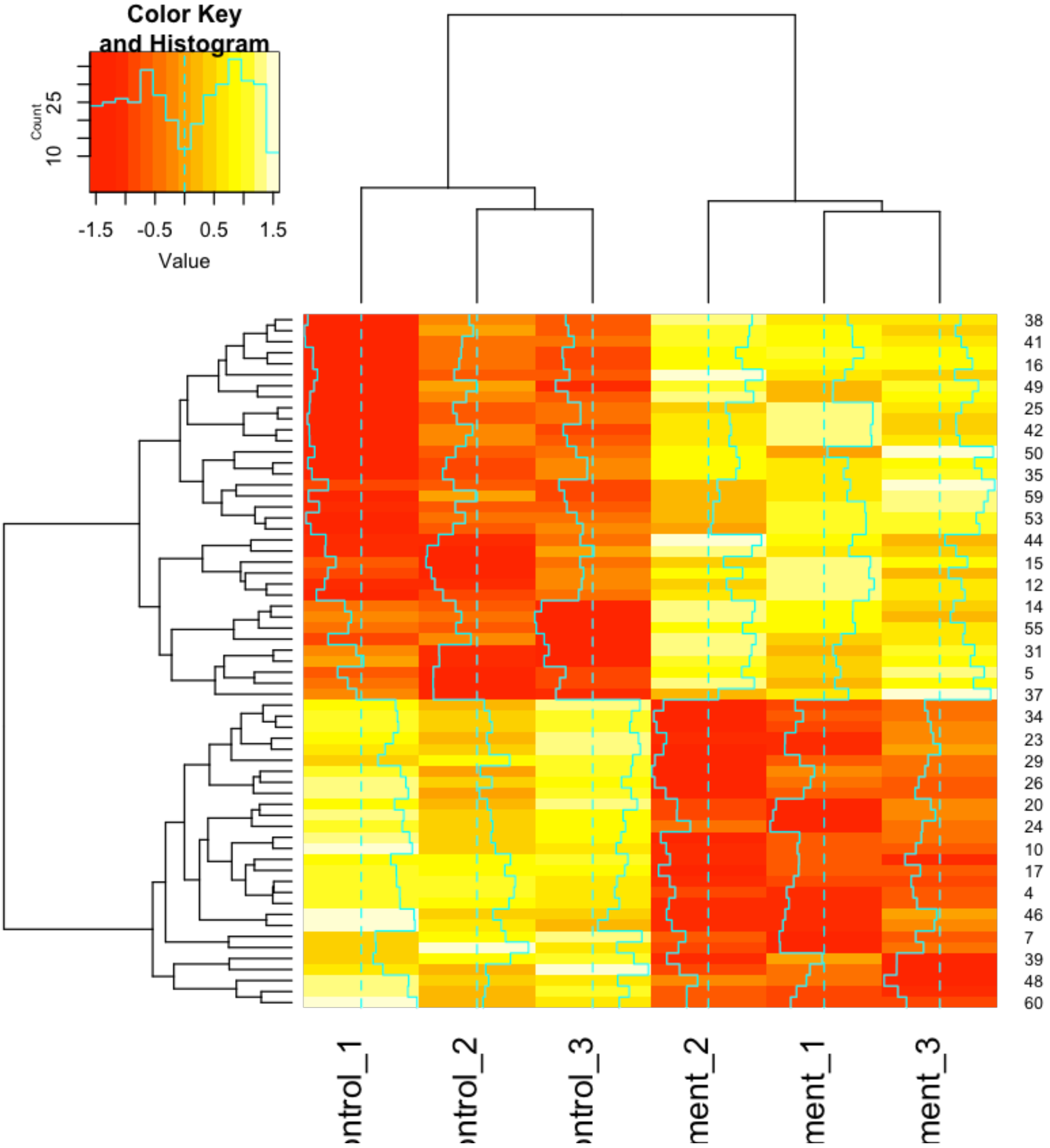
In [26]:

```
head(data_scaled)
```

Control_1	Control_2	Control_3	Treatment_1	Treatment_2	Treatment_3
0.9762782	1.0573011	0.6698701	-0.8391525	-1.0705155	-0.7937815
1.0708458	0.8831183	0.7383034	-0.9725425	-1.0876054	-0.6321195
0.8135388	0.9290403	0.9727051	-0.6946356	-1.0595914	-0.9610571
1.0649676	1.0211340	0.6039172	-1.0369089	-0.9183348	-0.7347751
-0.6521138	-1.2065806	-0.7467465	0.4593890	0.9189592	1.2270926
-1.2421272	-0.9388247	-0.3888625	1.2411314	0.6378125	0.6908705

In [27]:

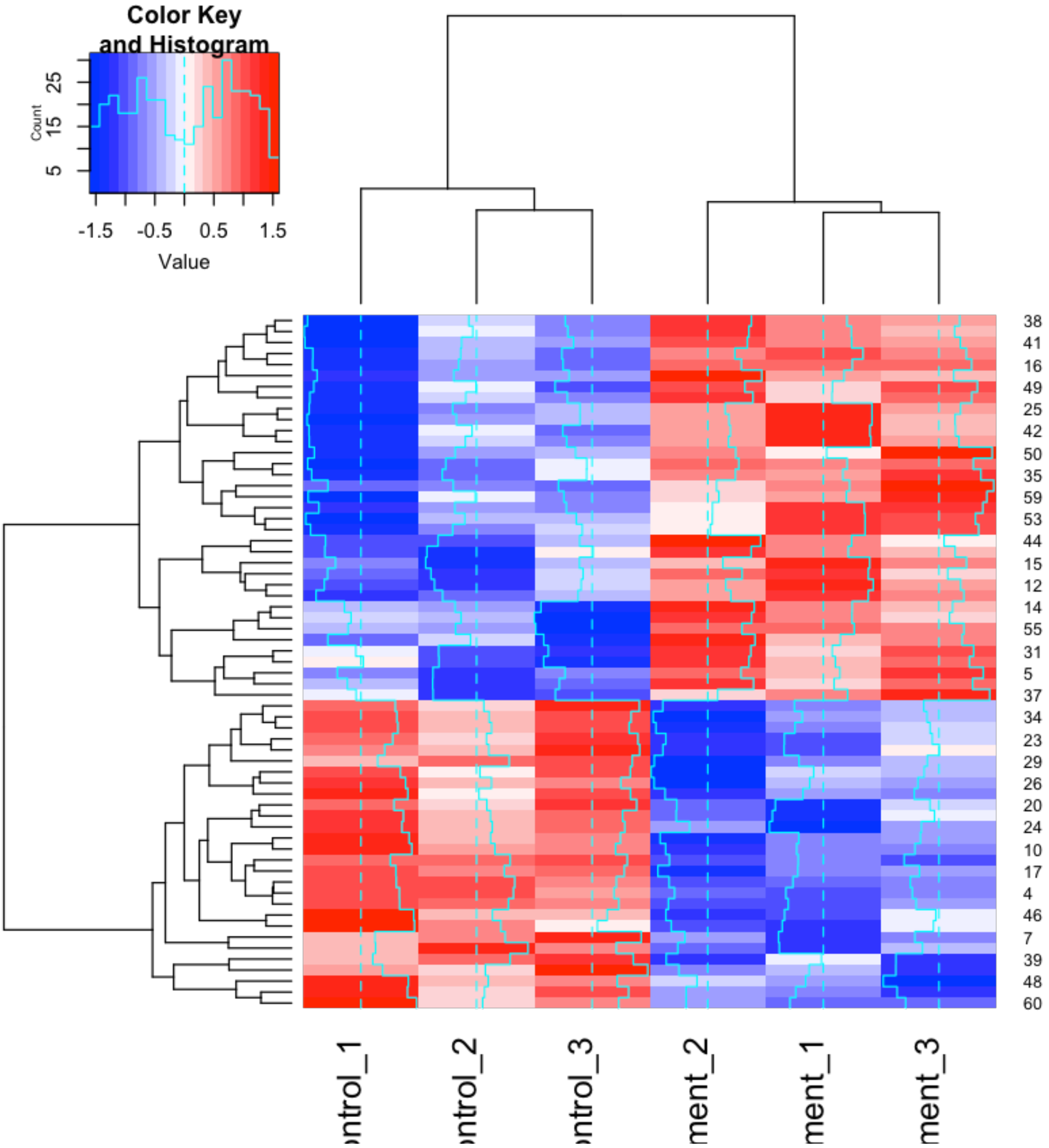
```
heatmap.2(data_scaled)
```



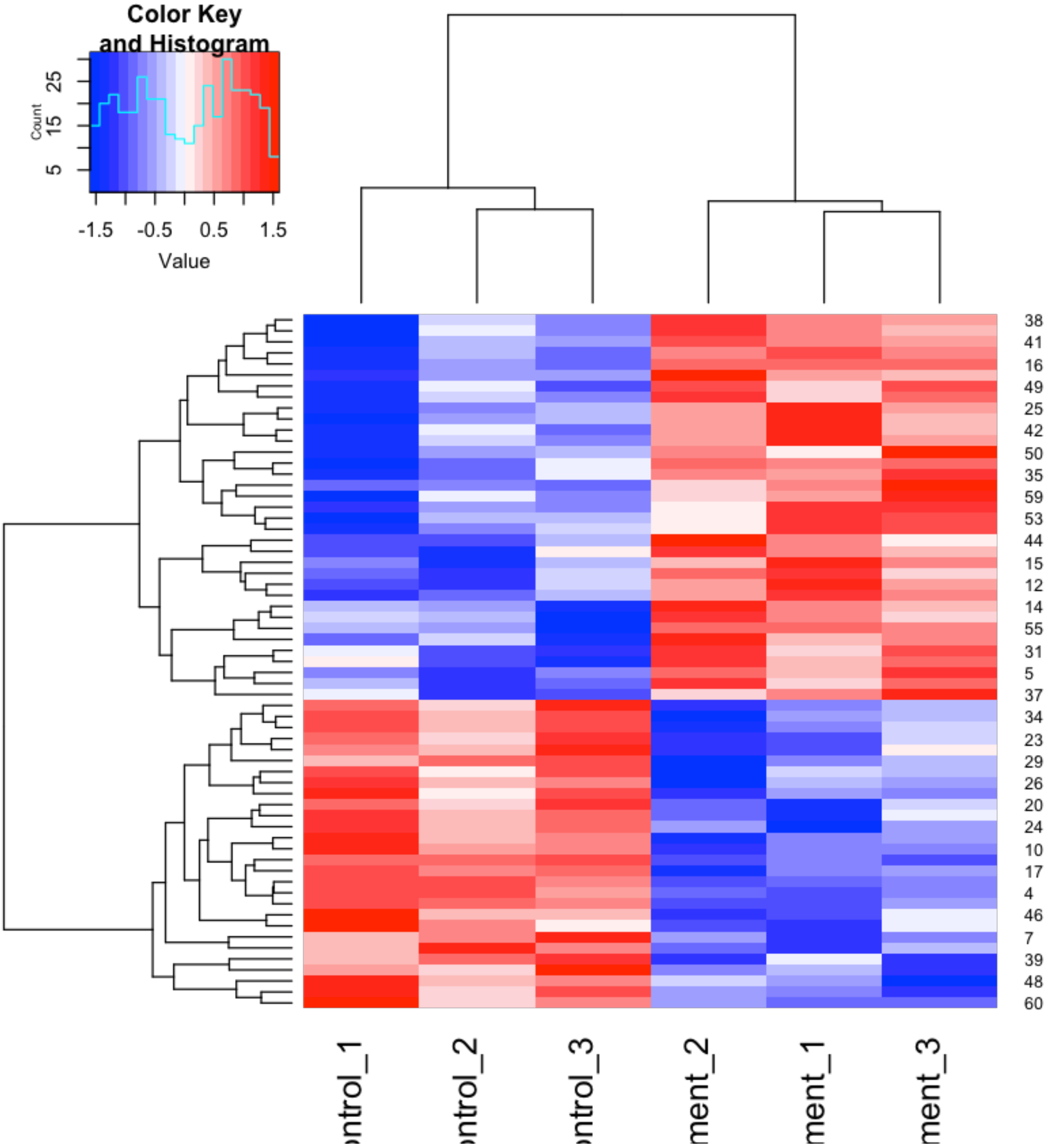
more control over colors

In [28]:
```
my_palette=colorRampPalette(c("blue",
"white","red"))
```

In [29]:
```
heatmap.2(data_scaled,
          col= my_palette(n=20))
```



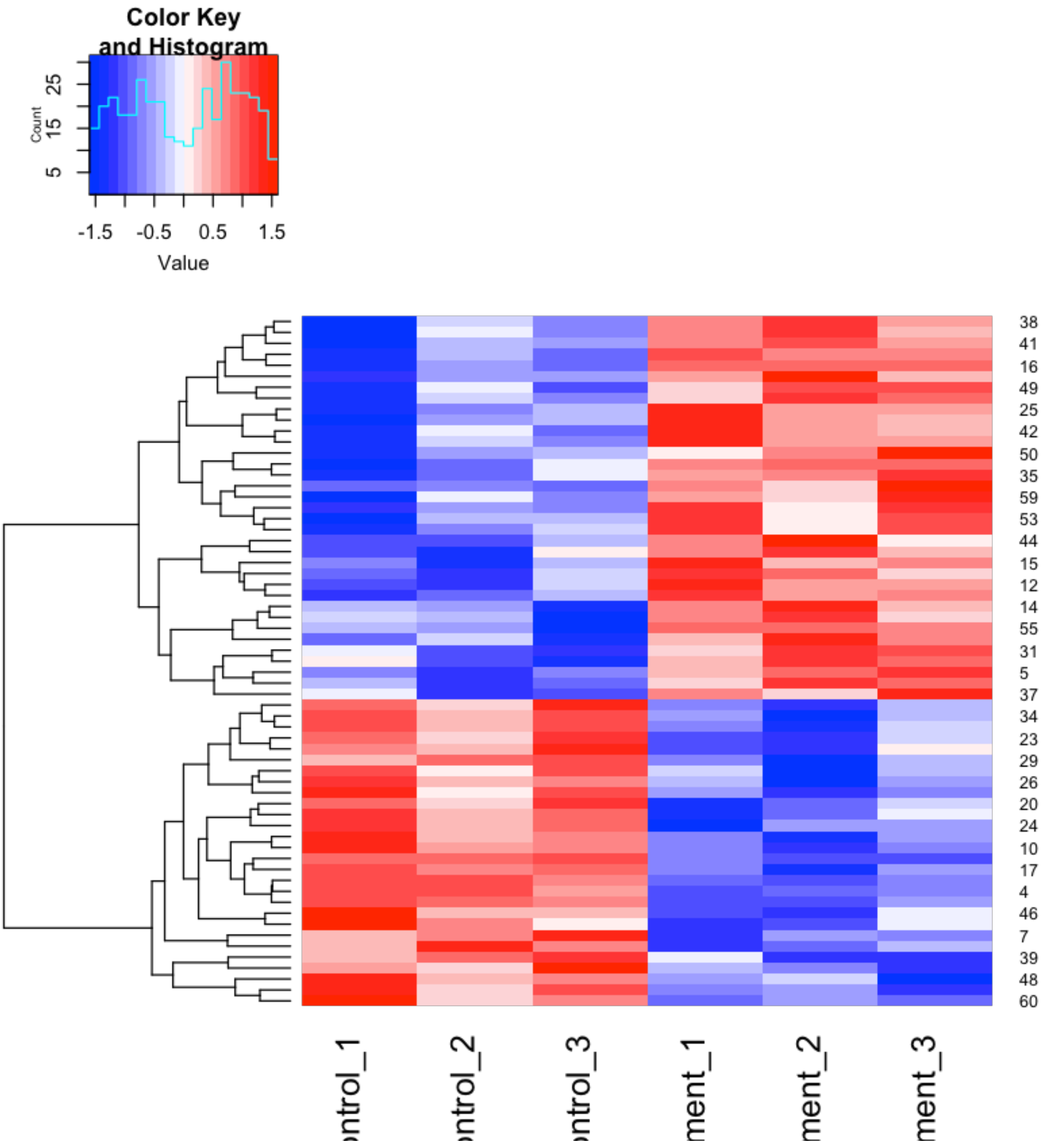
```
In [30]: heatmap.2(data_scaled,
                  col= my_palette(n=20),
                  trace = "none")
```



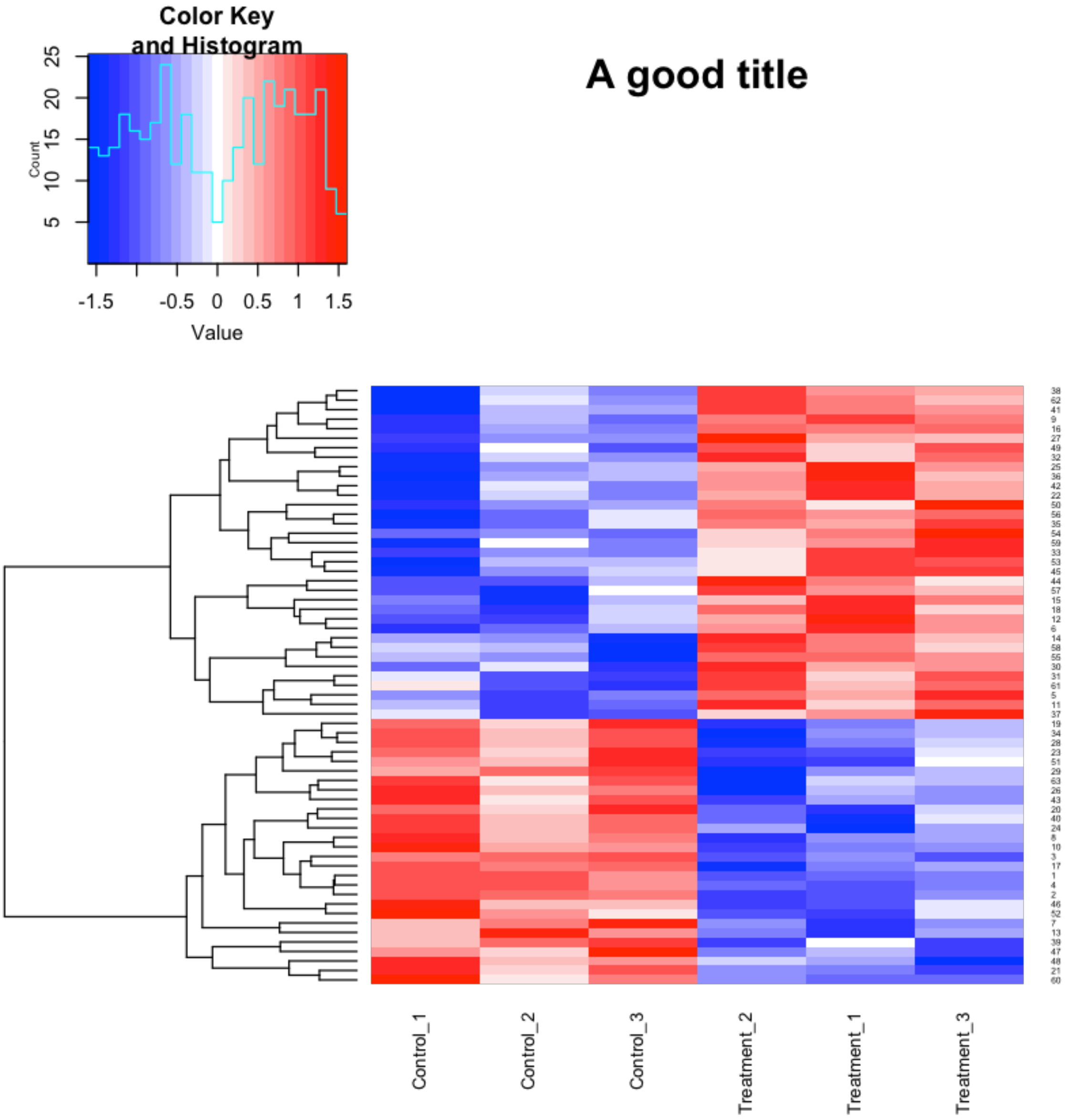
TO REMOVE THE CLUSTERS ON THE TOP

In [31]:
```
heatmap.2(data_scaled,
          col= my_palette(n=20),
          trace = "none",
          Colv = NA)
```

```
Warning message in heatmap.2(data_scaled, col = my_palette(n = 20)
, trace = "none", :
“Discrepancy: Colv is FALSE, while dendrogram is `both'. Omitting
column dendogram.”
```



In [52]:
```
heatmap.2(data_scaled,
dendrogram = "row", # only show the row dendrogram
trace = "none",
col = my_palette(25),
main = "A good title",
margins = c(6, 4),
keysize = 2,
cexRow = 0.40,
cexCol = 0.80)
```



The graph can be decorated according to our needs

Method 3 : pheat map

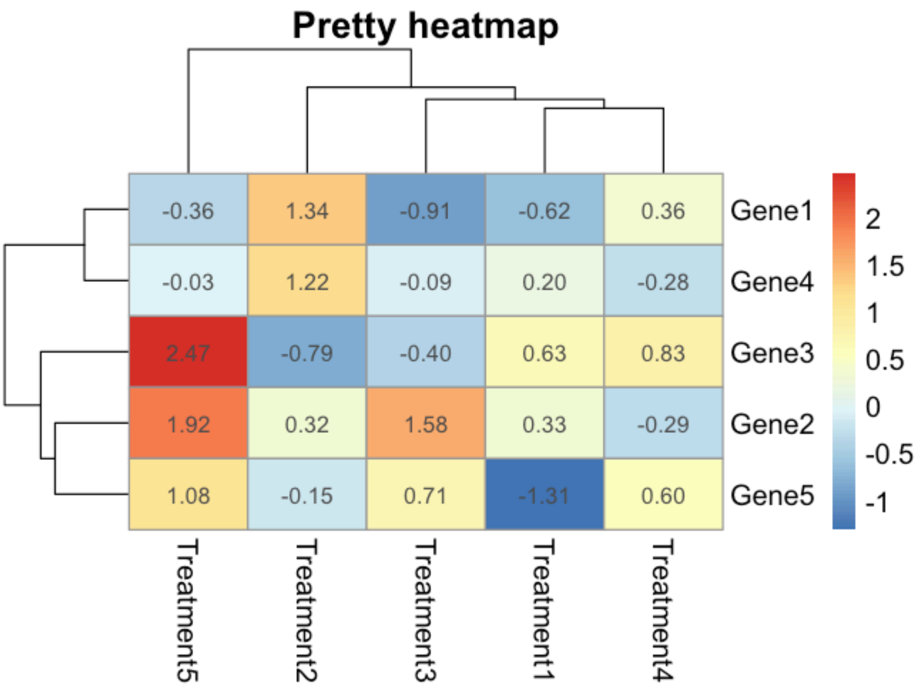
```
In [34]: install.packages("pheatmap")
```

```
Updating HTML index of packages in '.Library'
Making 'packages.html' ... done
```

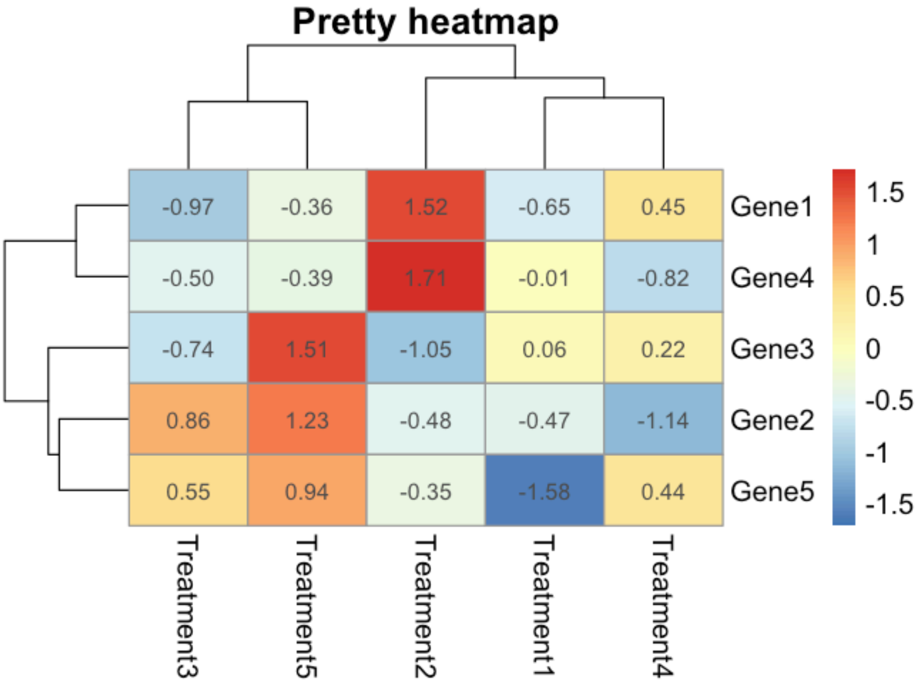
```
In [36]: library(pheatmap)
```

```
In [37]:  d <- matrix(rnorm(25), 5, 5)
          colnames(d) <- paste0("Treatment", 1:5)
          rownames(d) <- paste0("Gene", 1:5)
```

```
In [38]:  pheatmap(d,
          main = "Pretty heatmap",
          cellwidth = 50,
          cellheight = 30,
          fontsize = 12,
          display_numbers = TRUE)
```



```
In [48]: pheatmap(d,
          main = "Pretty heatmap",
          cellwidth = 50,
          cellheight = 30,
          fontsize = 12,
          display_numbers = TRUE,
          scale = "row")
```



	Treatment3	Treatment5	Treatment2	Treatment1	Treatment4
Gene1	-0.97	-0.36	1.52	-0.65	0.45
Gene4	-0.50	-0.39	1.71	-0.01	-0.82
Gene3	-0.74	1.51	-1.05	0.06	0.22
Gene2	0.86	1.23	-0.48	-0.47	-1.14
Gene5	0.55	0.94	-0.35	-1.58	0.44

Numbers can also be viewed in the graph by giving display Numbers = "TRUE"

In [49]:

```
pheatmap(d,
main = "Pretty heatmap",
cellwidth = 50,
cellheight = 30,
fontsize = 12,
filename = "/Users/sanju/Desktop/R Programming/heatmap.pdf")
```

We can save the heat map in a pdf fromat in our system

The summary of the 3 methods

In [42]:

```

#stats::heatmap(): scale (row) -> cluster -> colour
#gplots::heatmap.2(): cluster -> scale (none) -> colour
#pheatmap::pheatmap(): scale (none) -> cluster -> colour
```

Method 4 : A dataframe in ggplot2

In [43]:

```
library(ggplot2)
```

```
Registered S3 methods overwritten by 'ggplot2':
  method         from 
  [.quosures     rlang
  c.quosures     rlang
  print.quosures rlang
```

Subsetting the data we want to look at only the age of 55 to 64

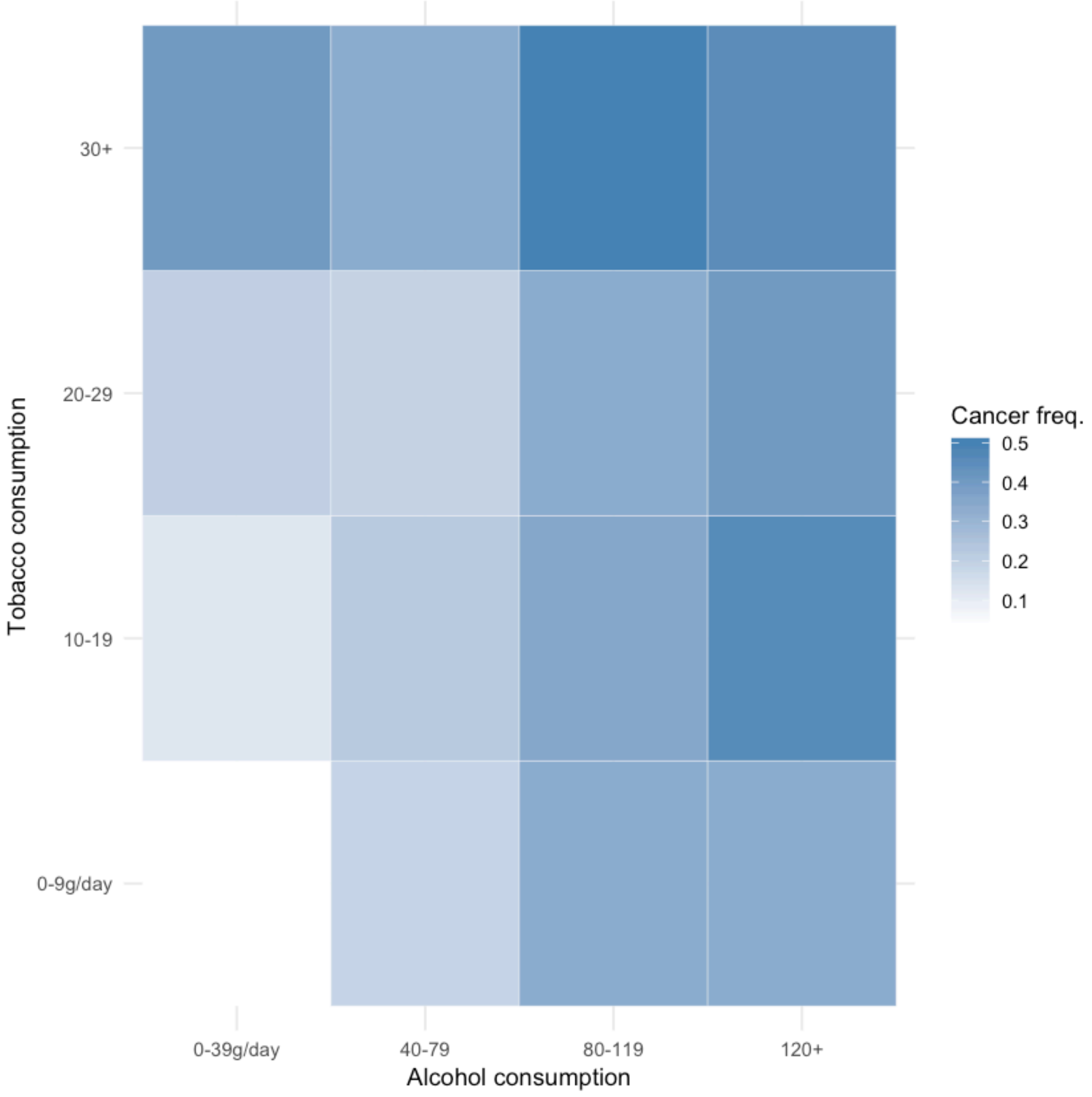
In [44]:

```
esoph_sub <- subset(esoph, agegp == "55-64")
```

creating a basic heatmap from the data frame

In [45]:

```
ggplot(esoph_sub, aes(x = alcgp,
                      y = tobgp,
                       fill = ncases / (ncases + ncontrols))) +
geom_tile(colour = "white") + # grid colour
scale_fill_gradient(low = "white",
high = "steelblue") +
theme_minimal() +
labs(fill = "Cancer freq.",
x = "Alcohol consumption",
      y = "Tobacco consumption")
```



This method will not allow to create any dendrograms

In [ ]: