

Introduction and overview

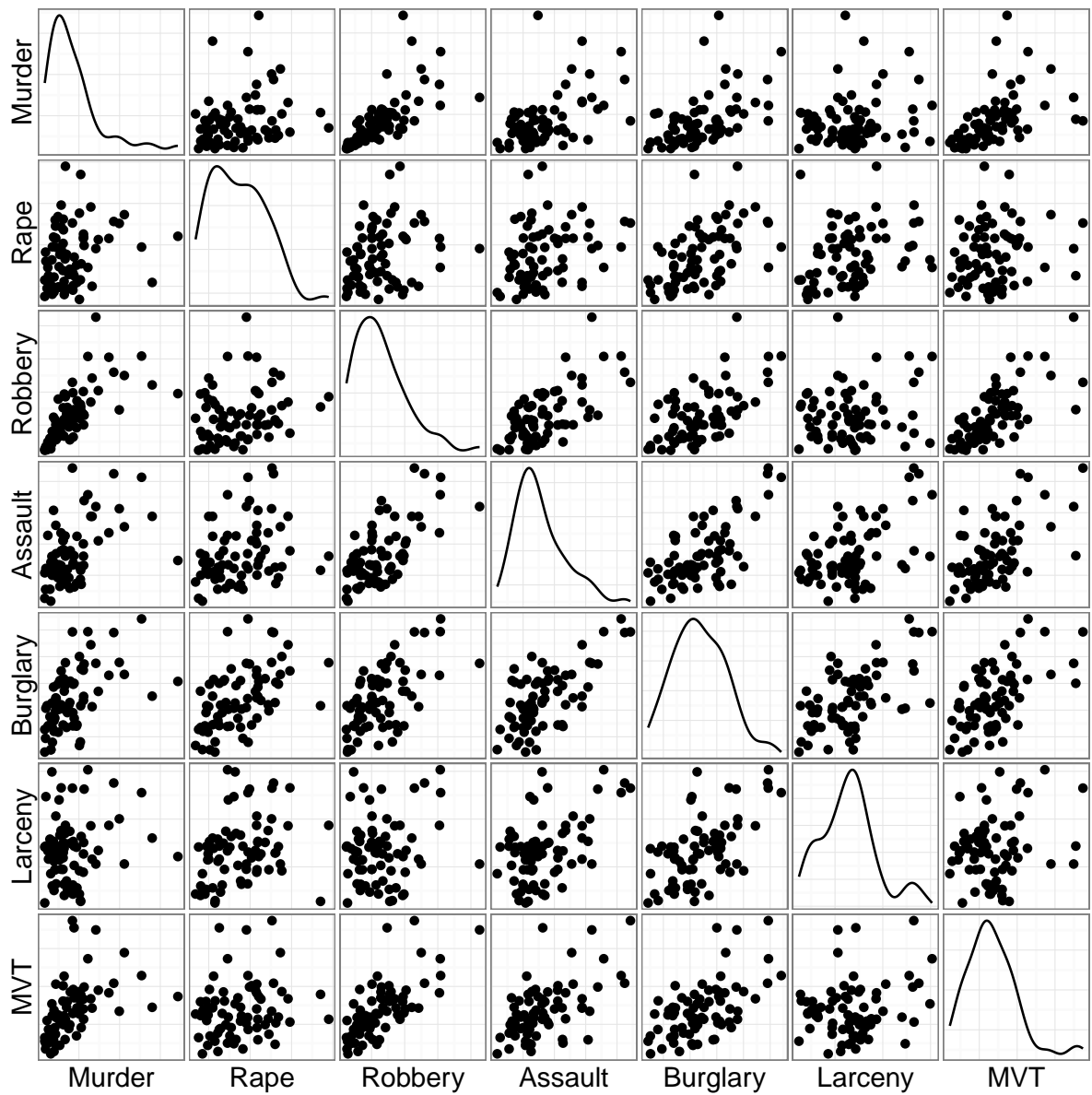
STATS 503

Here is the code to create the plots from the lecture slides. The datasets are located on CTools.

1 Crime dataset

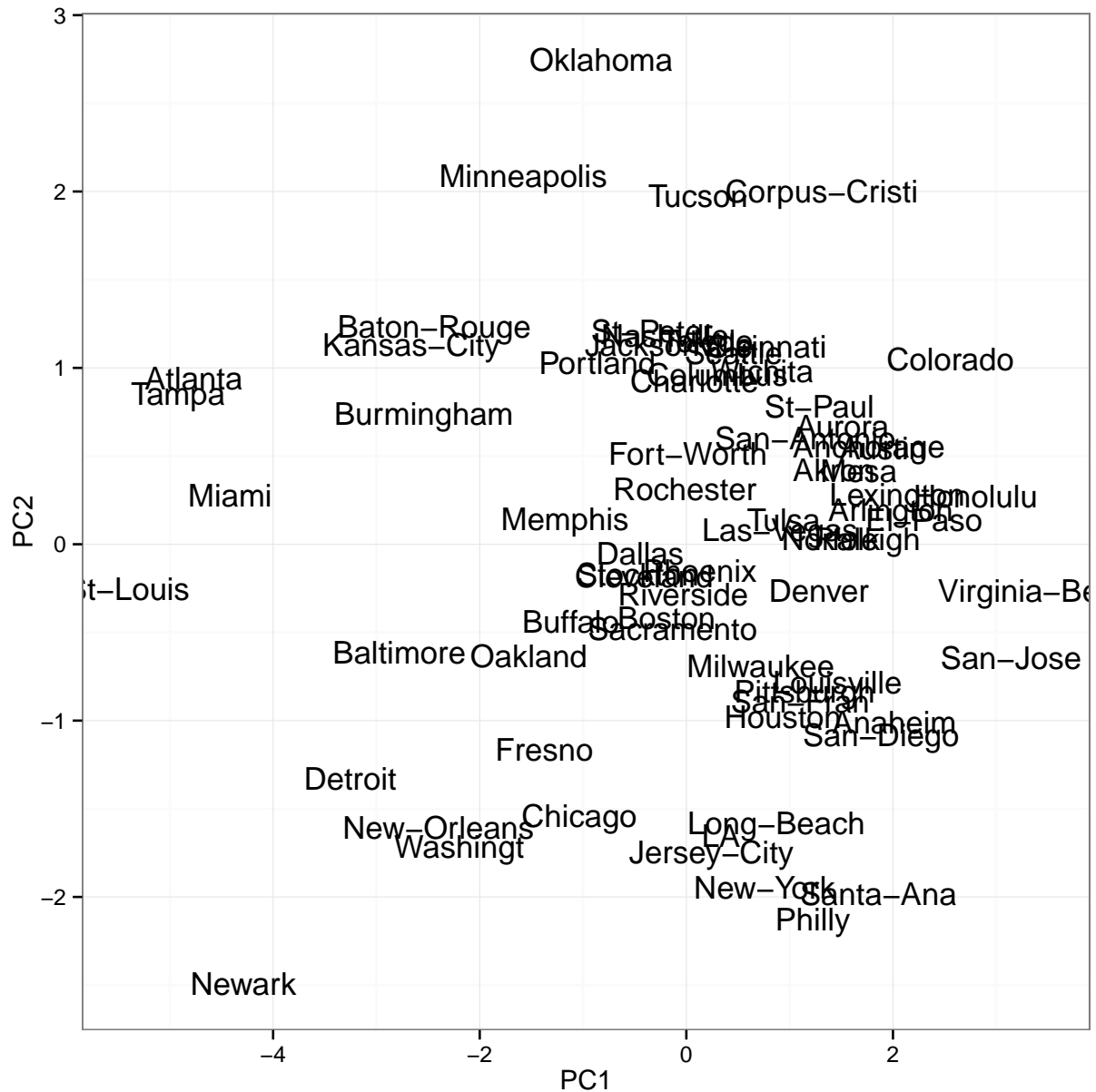
First, the scatterplot of crime data using `ggpairs`.

```
#install.packages("ggplot2")
#install.packages("GGally")
library(ggplot2)
library(GGally)
data_crime <- read.table("citycrime.txt", header = TRUE)
ggpairs(data_crime, axisLabels = "none",
        upper = list(continuous = "points", combo = "dot"),
        lower = list(continuous = "points", combo = "dot")) +
  theme_bw()
```



Now, principal component analysis is performed and the cities are projected in a plane.

```
city.PCAcor <- princomp(data_crime,cor=T)
cityPC = predict(city.PCAcor)
qplot(x = cityPC[,1],y=cityPC[,2],label=row.names(data_crime),
      geom = "text",xlab = "PC1", ylab = "PC2")+theme_bw()
```



2 Sleeping bag data

To create a mosaic plot, we use the next function.

```
## Stackoverflow:
##http://stackoverflow.com/questions/19233365/how-to-create-a-marimekko-
##mosaic-plot-in-ggplot2
ggMMplot <- function(var1, var2){
  require(ggplot2)
  levVar1 <- length(levels(var1))
  levVar2 <- length(levels(var2))

  jointTable <- prop.table(table(var1, var2))
```

```

plotData <- as.data.frame(jointTable)
plotData$marginVar1 <- prop.table(table(var1))
plotData$var2Height <- plotData$Freq / plotData$marginVar1
plotData$var1Center <- c(0, cumsum(plotData$marginVar1)[1:levVar1 -1]) +
  plotData$marginVar1 / 2

ggplot(plotData, aes(var1Center, var2Height)) +
  geom_bar(stat = "identity", aes(width = marginVar1,
                                   fill = var2), col = "Black") +
  geom_text(aes(label = as.character(var1), x = var1Center, y = 1.05)) +
  theme(axis.ticks = element_blank(), axis.text.x = element_blank(),
        axis.text.y = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_blank())
}

```

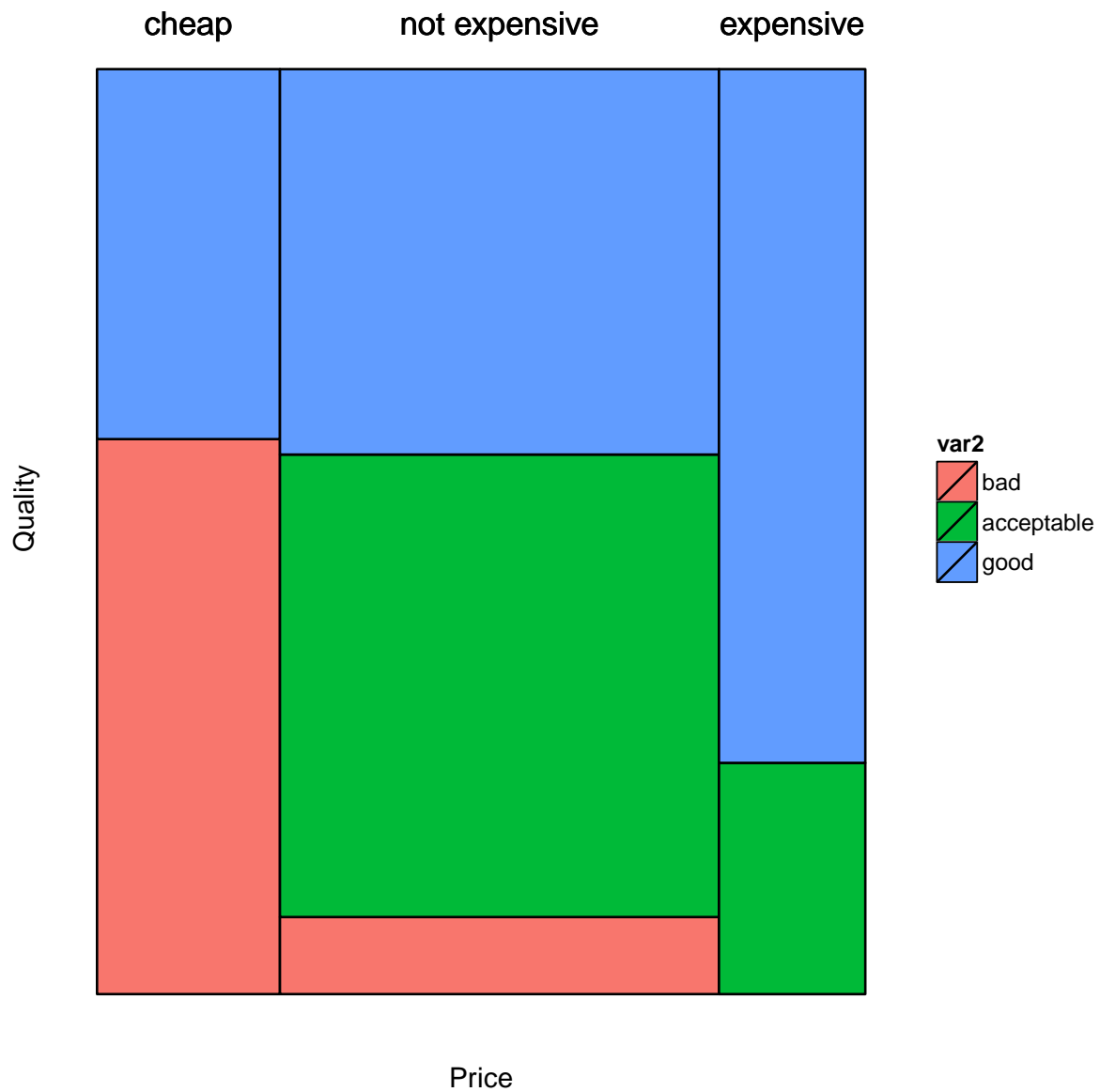
Now, load the data and construct a table to create a mosaic plot.

```

data = read.table("sleeping_bags.txt")
sleeping_table = table(data[,c(1,3)])
# Reorder categories
sleeping_table = as.table(sleeping_table[c(1,3,2),c(2,1,3)])

df = as.data.frame(sleeping_table)
ggMplot(var1 = rep(df$price,df$Freq),var2 = rep(df$quality,df$Freq)) +
  xlab("Price")+ylab("Quality")

```



To project the data in two dimensions, perform multiple correspondence analysis (MCA).

```
library(FactoMineR)
sleeping.mca = MCA(data, graph = FALSE)

vars_coord = data.frame(sleeping.mca$var$coord)
obs_coord = data.frame(sleeping.mca$ind$coord)

# Coordinates of lines
x = rep(NA, nrow(obs_coord)*ncol(data)*2)
x[0:(nrow(obs_coord)*ncol(data)-1)*2+1] = rep(obs_coord[,1], each = 3)
x[0:(nrow(obs_coord)-1)*2*ncol(data)+2] = vars_coord[as.character(data[,1]),1]
x[0:(nrow(obs_coord)-1)*2*ncol(data)+4] = vars_coord[as.character(data[,2]),1]
x[0:(nrow(obs_coord)-1)*2*ncol(data)+6] = vars_coord[as.character(data[,3]),1]
```

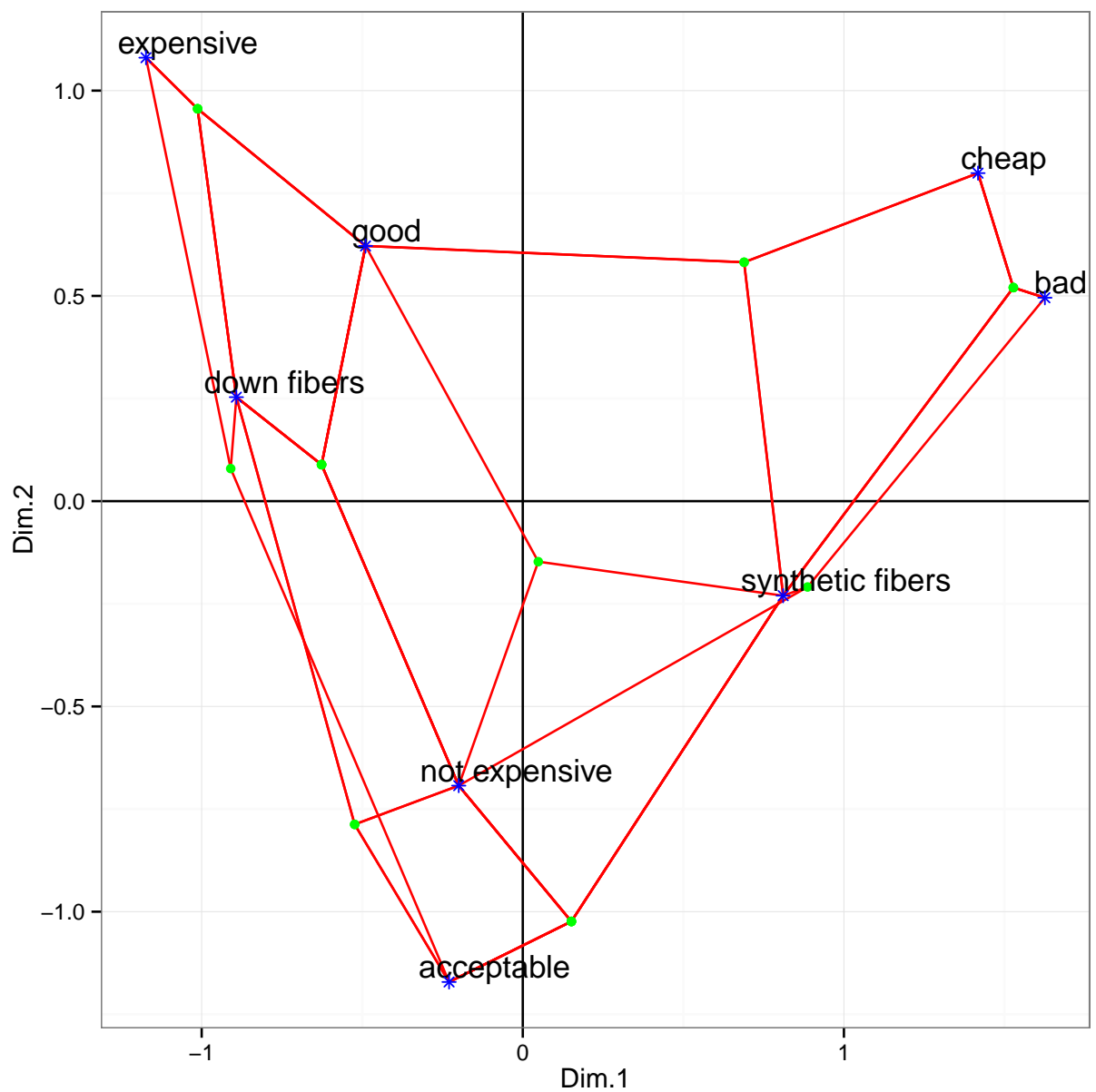
```

y = rep(NA, nrow(obs_coord)*ncol(data)*2)
y[0:(nrow(obs_coord)*ncol(data)-1)*2+1] = rep(obs_coord[,2],each = 3)
y[0:(nrow(obs_coord)-1)*2*ncol(data)+2] = vars_coord[as.character(data[,1]),2]
y[0:(nrow(obs_coord)-1)*2*ncol(data)+4] = vars_coord[as.character(data[,2]),2]
y[0:(nrow(obs_coord)-1)*2*ncol(data)+6] = vars_coord[as.character(data[,3]),2]

line = rep(1:(nrow(obs_coord)*ncol(data)),each = 2)
xy_df = data.frame(x,y,line)

#http://stackoverflow.com/questions/30603952/multiple-correspondence-analysis-in-r-plotting
# -supplementary-categorical-variab
ggplot(data = obs_coord, aes(x = Dim.1, y = Dim.2)) +
  geom_hline(yintercept = 0, colour = "black") +
  geom_vline(xintercept = 0, colour = "black") +
  geom_line(data=xy_df, aes(x=x, y=y, group = line),colour = "red")+
  geom_point(colour = "green", alpha = 1,sahpe = 19) +
  geom_text(data = vars_coord,
            aes(x = Dim.1, y = Dim.2,
                label = rownames(vars_coord)),hjust=0.2, vjust=-0.2) +
  geom_point(data = vars_coord, colour ="blue",shape = 8) +theme_bw()

```



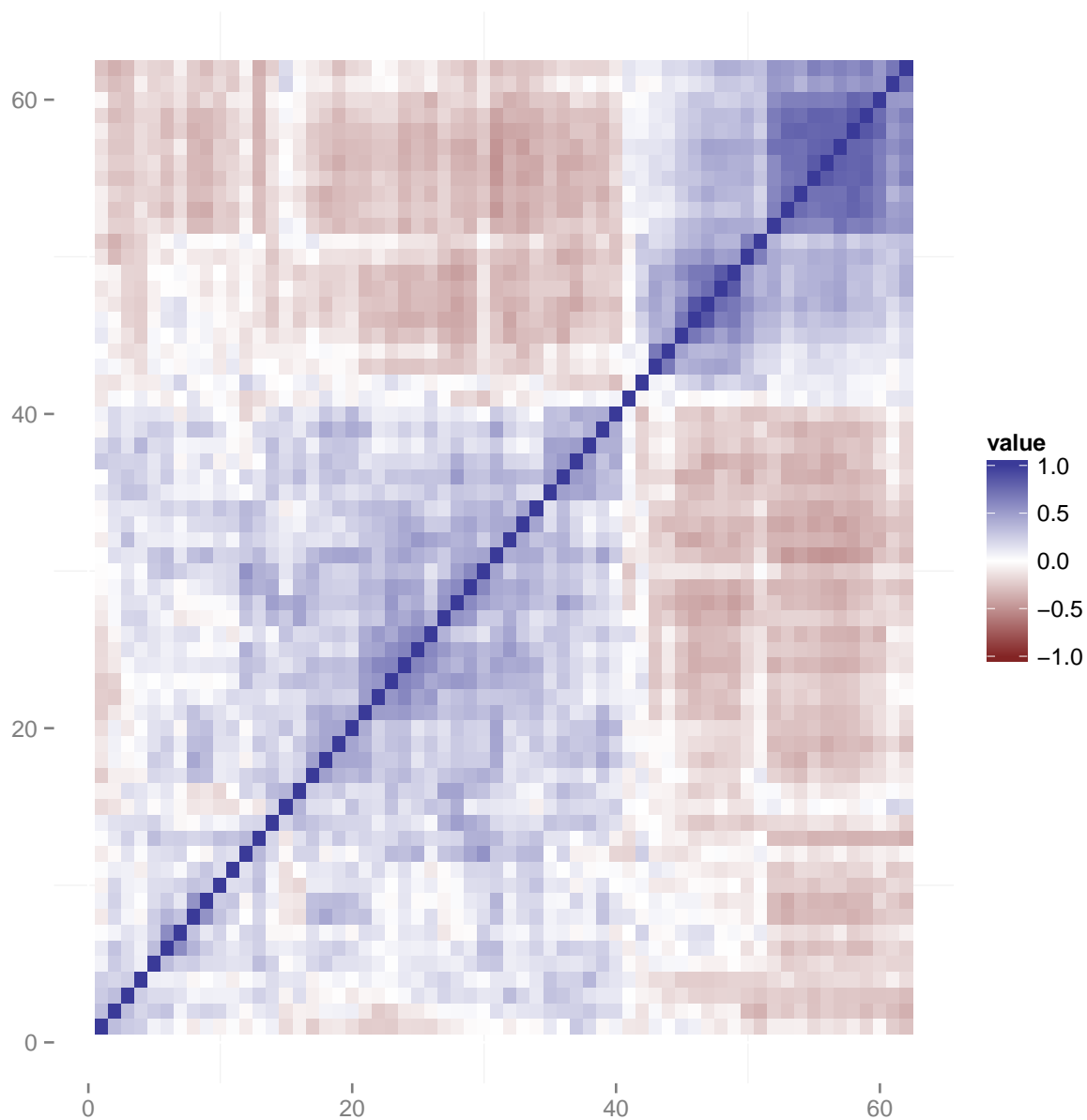
3 Lymphona data

The lymphona data can be loaded from the library `spls`. Then, plot a heatmap of the correlation matrix.

```
#install.packages("spls")
library(spls)
data(lymphoma)
lymph.cor = cor(t(lymphoma$x))

qplot(x=Var1, y=Var2, data=melt(lymph.cor), fill=value, geom="tile") +
  scale_fill_gradient2(limits=c(-1, 1)) +
  theme(axis.title.x=element_blank(),
        axis.title.y=element_blank(),
```

```
panel.background = element_blank()
```



To embed the data in two dimensions, multidimensional scaling (MDS) is performed.

```
lymph.mds = cmdscale(1-lymph.cor, k=2)  
  
qplot(x = lymph.mds[,1], y = lymph.mds[,2], geom = "text", label = 1:62,  
      xlab = "MDS1", ylab = "MDS2")
```