# Density Estimation 2

GMM. DBSCAN

Caballero Vergés Biel, Menzenbach Svenja and Reyes Illescas Kleber Enrique

2023-10-11

```
load("BikeDay.Rdata")
X <- as.matrix(day[day$yr==1,c(10,14)])
```

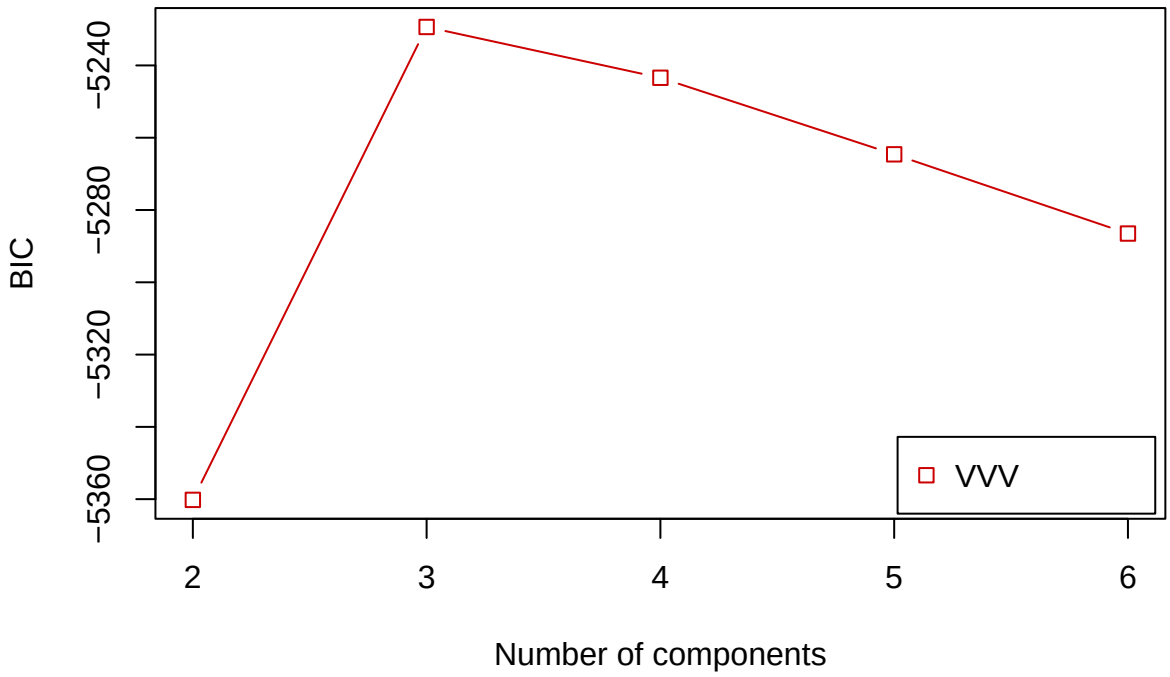
## Questions

### 1.

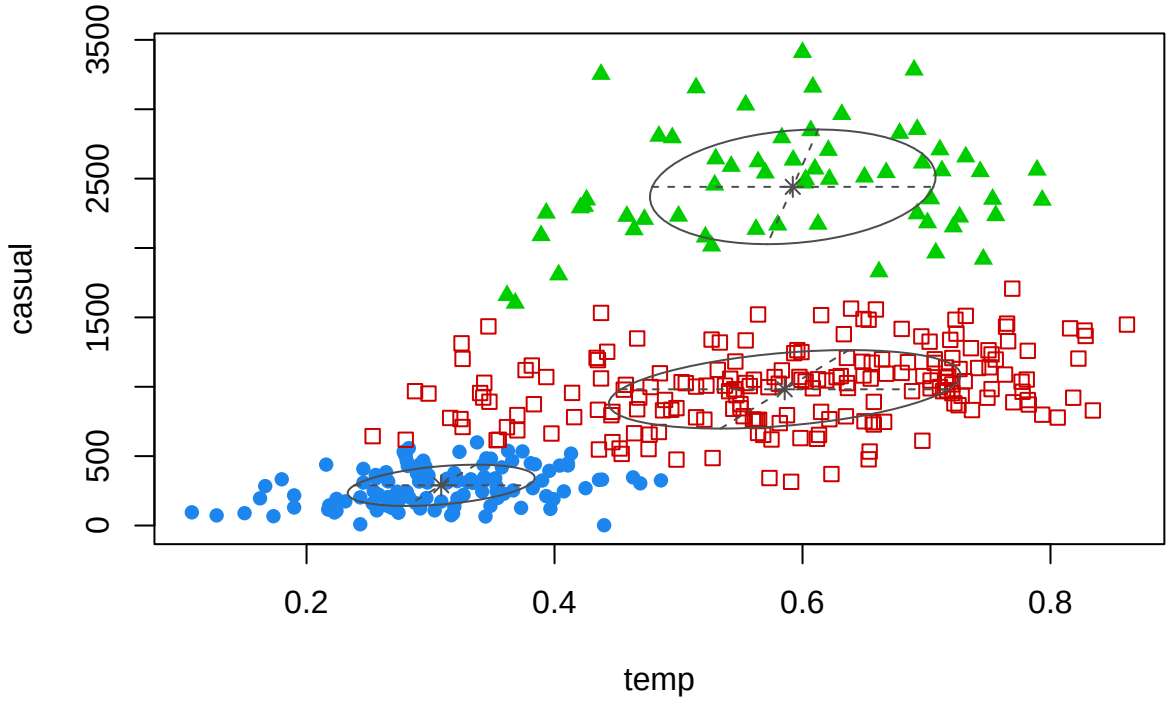
```
GMM_BIC <- Mclust(X,G=2:6, modelNames="VVV")
summary(GMM_BIC, parameters=F)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VVV (ellipsoidal, varying volume, shape, and orientation) model with 3
## components:
##
##  log-likelihood   n df       BIC       ICL
##       -2564.509 366 17 -5229.362 -5261.588
##
## Clustering table:
##   1   2   3
## 111 195  60
```

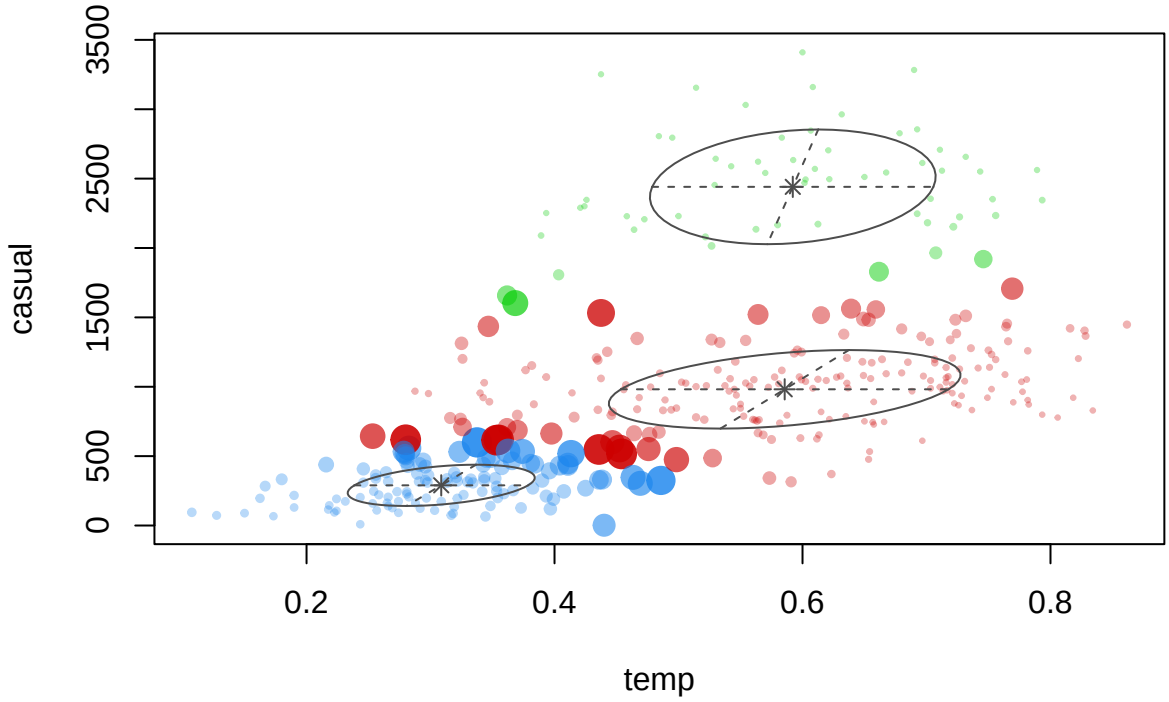
```
plot(GMM_BIC, what="BIC")
```



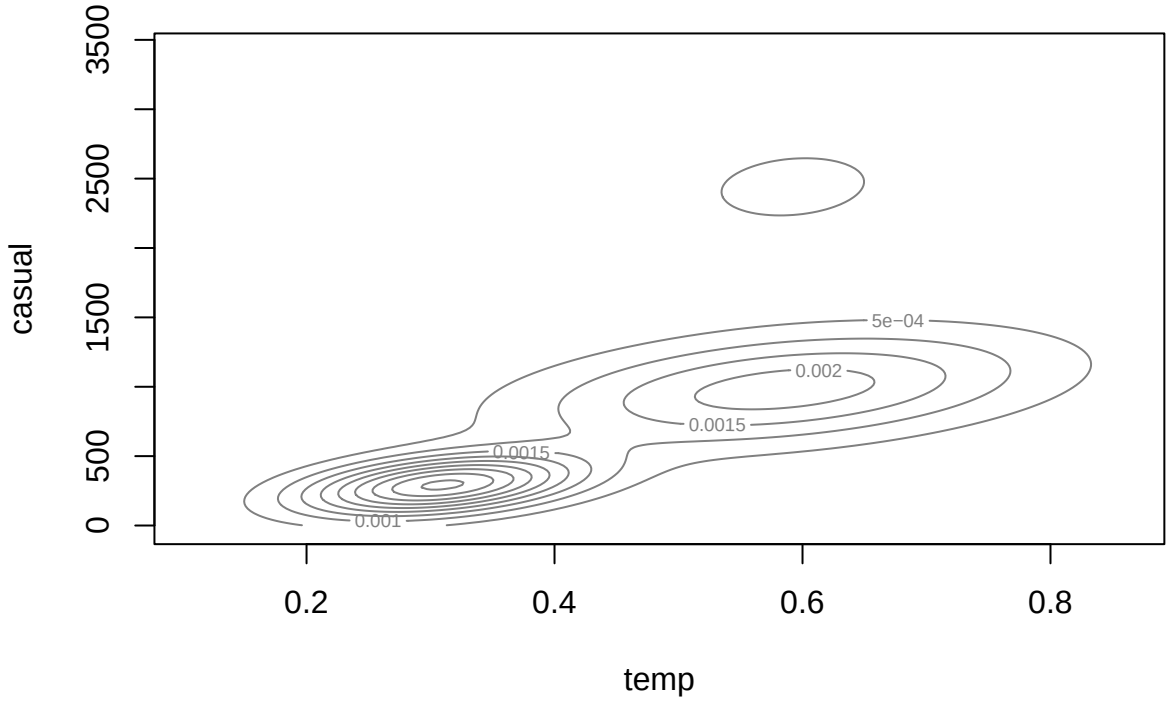

```
plot(GMM_BIC, what="classification")
```




```
plot(GMM_BIC, what="uncertainty")
```




```
plot(GMM_BIC, what="density")
```

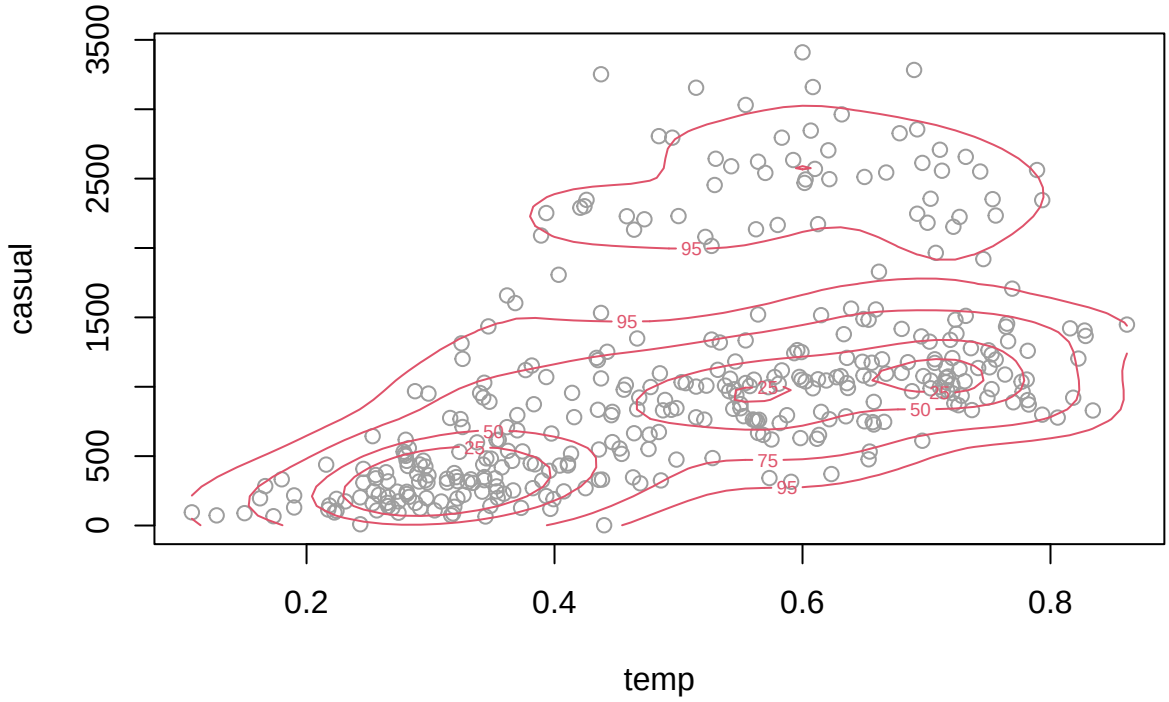



According to BIC, the best number of clusters is 3. It is the one that minimizes the BIC among the range of different k values we have given.

We can highlight the fact in the uncertainty plot, we have several points that are very large. This implies that the classification might not be accurate.

**2.**

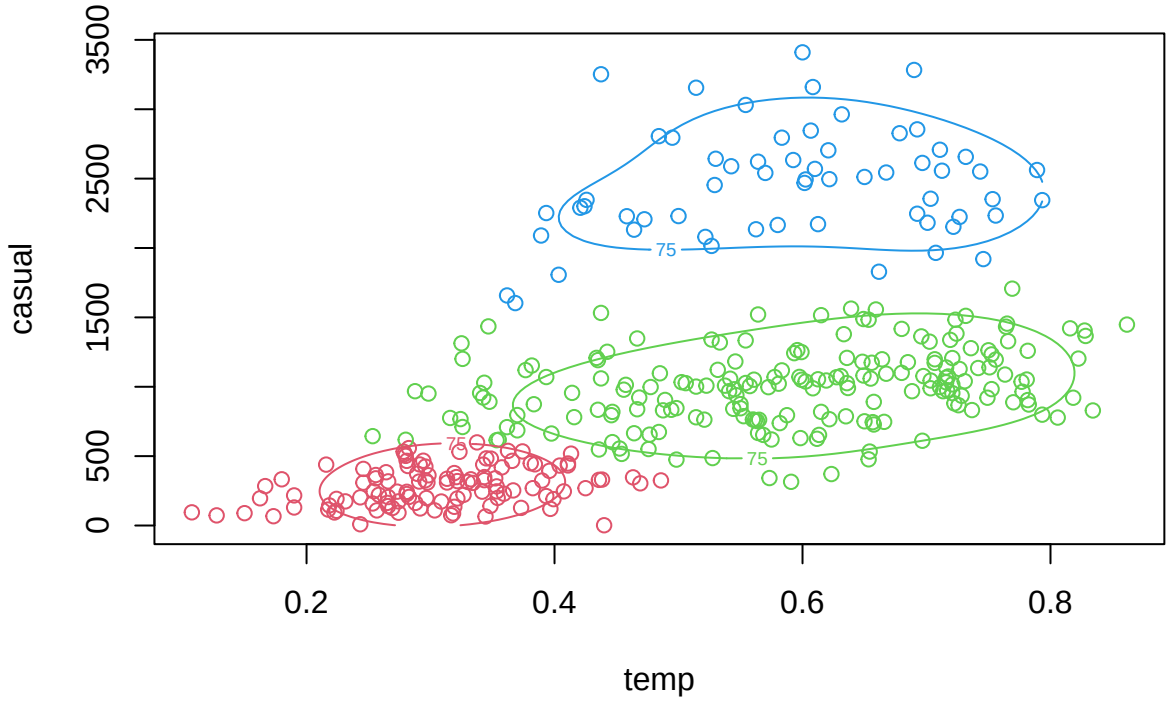
```
a = 0.25

plot(X, col=8)
sm.density(X,h=a*c(sd(day$temp), sd(day$casual)),
           display="slice",col=2, props=c(25,50,75,95), add=TRUE)
```




The previous density plot has perfect ellipsoid contour lines. This lines are approximately ellipsoid as well but less smooth. The distribution is very similar but this plot has one maximum more in the lower right cluster.

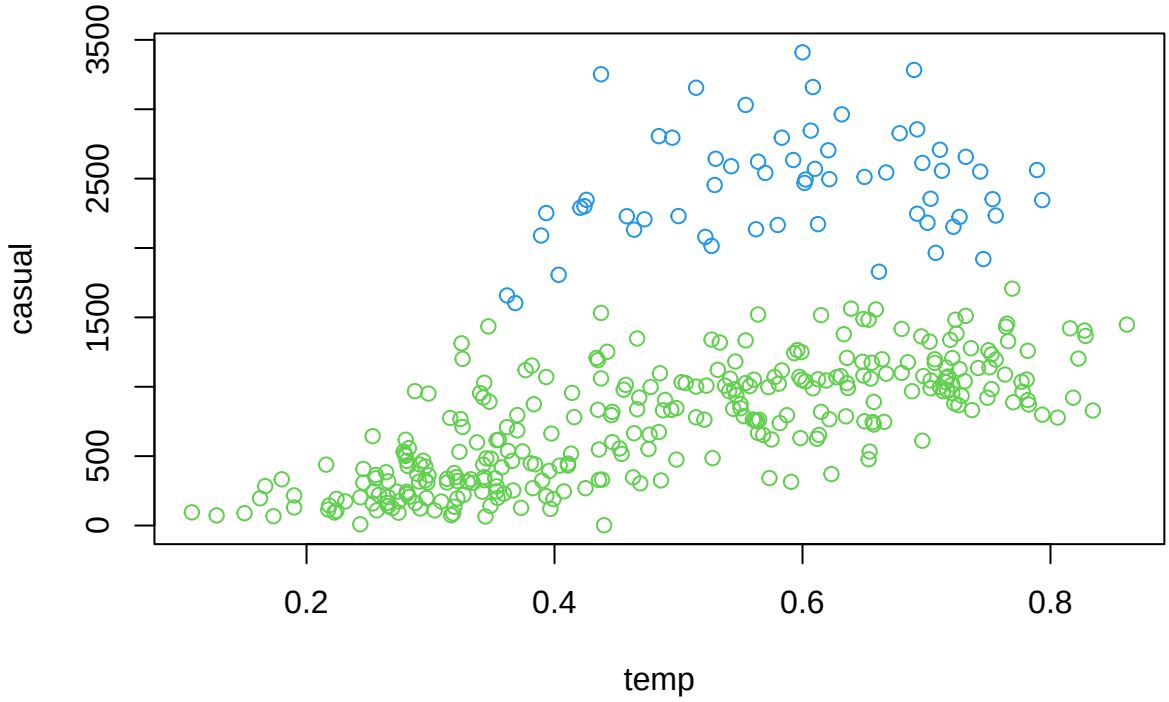
### 3.

```
k <- GMM_BIC$G
clust.ind <- GMM_BIC$classification

plot(X,col=1+clust.ind)

a = 0.4

for (j in 1:k){
  cl.j <- (clust.ind==j)
  sm.density(X[cl.j,],h=a*c(sd(day$temp), sd(day$casual)),
             display="slice",props=c(75),
             col=1+j, cex=4, add=TRUE)
}
```




**4.**

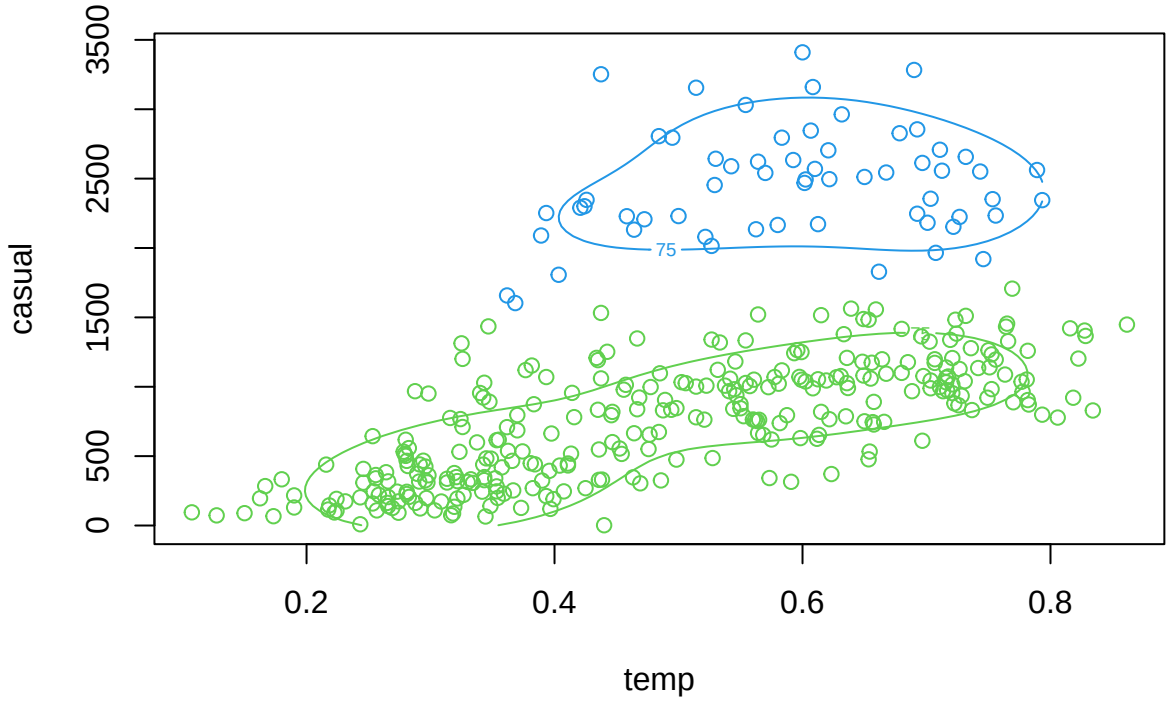
```
cm <- mclustBIC(X,G=2:6,modelNames="VVV")
cmnbhat <- mergenormals(X, summary(cm, X), method="bhat")

plot(X, col=2+cmnbhat$clustering)
```




In the previous exercise, we have the three different clusters, each with its respective density, in a single plot. Now, we have used mergenormal function in order to check if it's possible to merge some components. As we can see, effectively, the 2 lower clusters were merge to create a large one (in green).

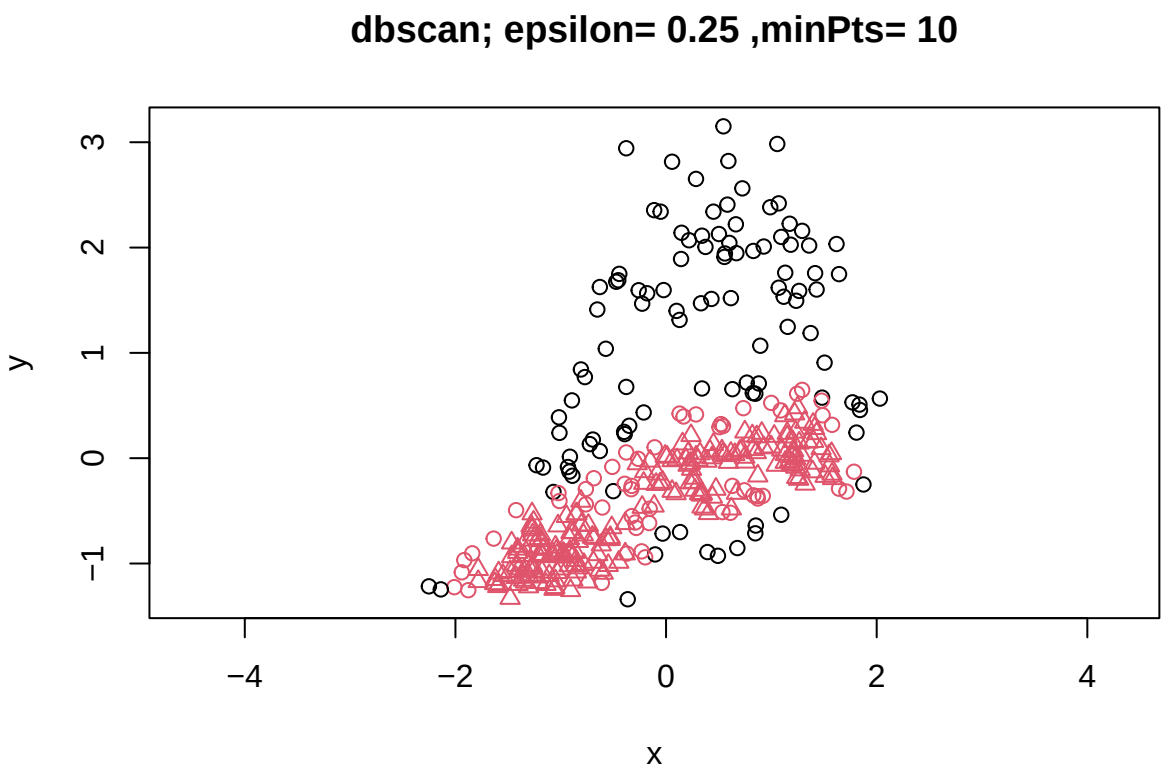
**5.**

```
k_new <- length(cmnbhat$clusternumbers)
clust_merged.ind <- cmnbhat$clustering

plot(X,col=2+clust_merged.ind)

a = 0.4

for (j in 1:k_new){
  cl.j <- (clust_merged.ind==j)
  sm.density(X[cl.j,],h=a*c(sd(day$temp), sd(day$casual)),
             display="slice",props=c(75),
             col=2+j, cex=4, add=TRUE)
}
```

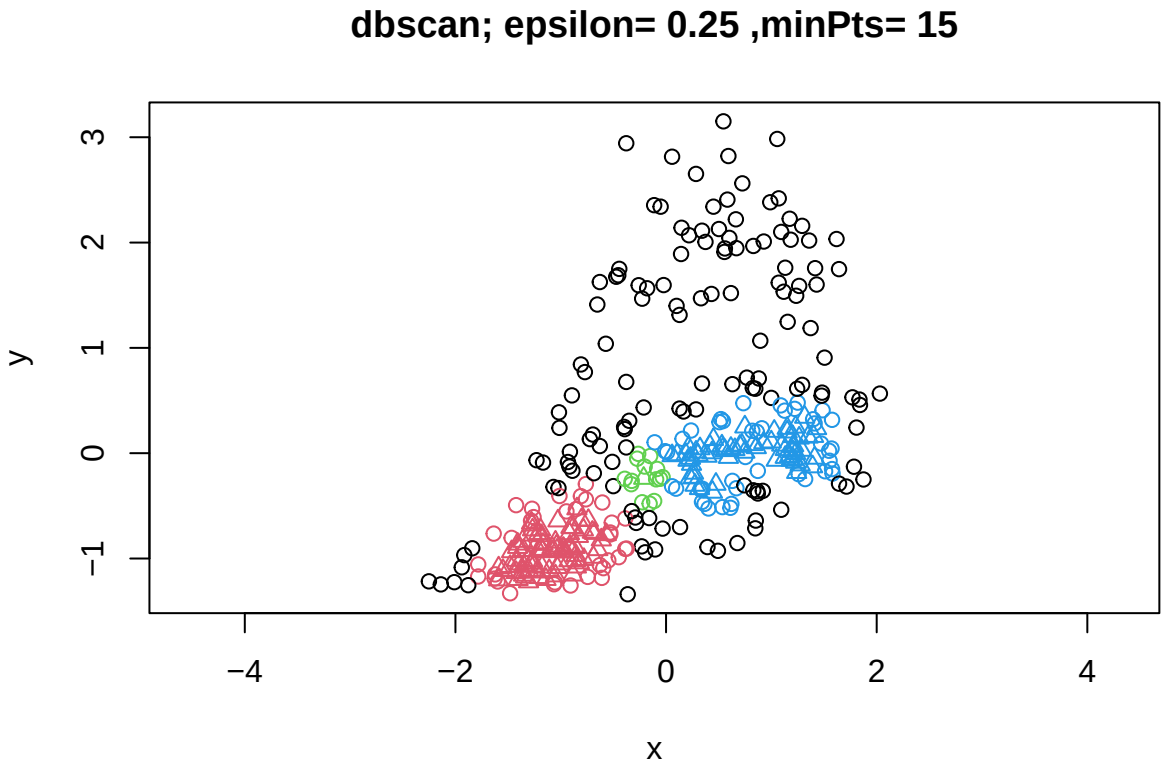



**6.**

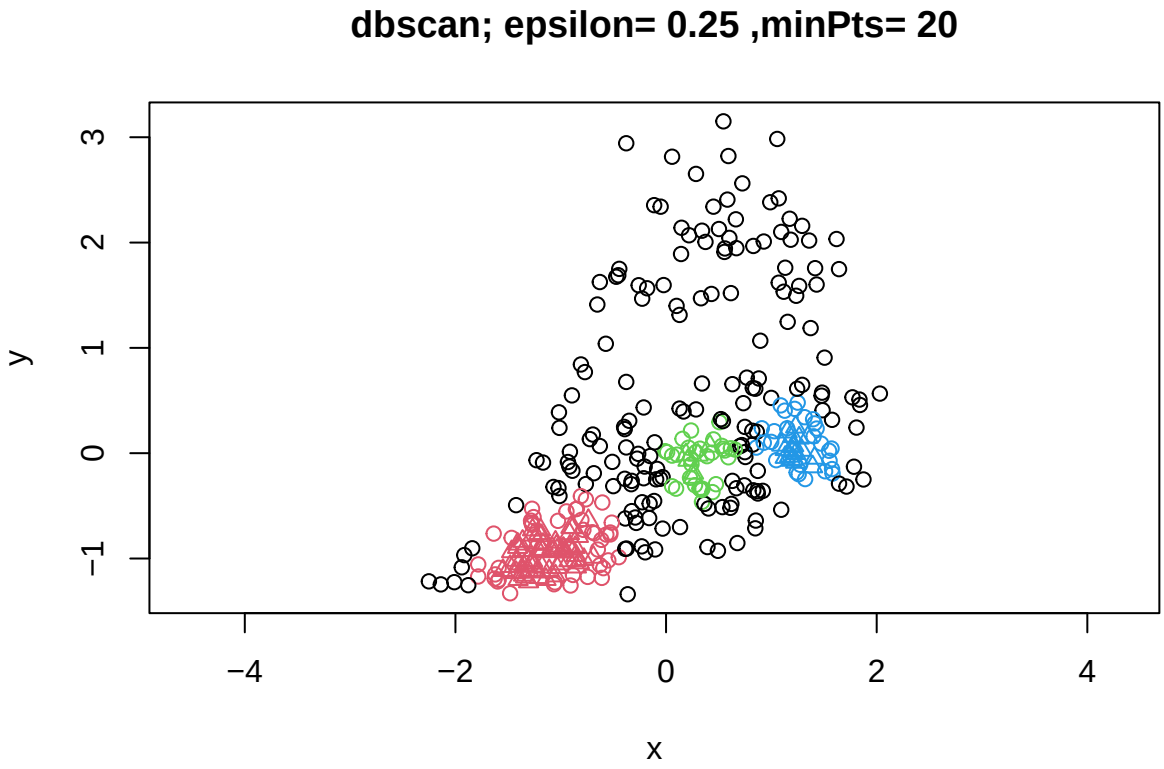
```
Xs <- scale(X)

eps_values <- c(0.25, 0.5)
minPts_values <- c(10, 15, 20)


for (eps in eps_values) {
  for (minPts in minPts_values) {

    dbscan_result <- fpc::dbscan(Xs, eps = eps, MinPts = minPts, showplot = 0)
    plot(dbscan_result, Xs, main=paste("dbscan; epsilon=",eps,",minPts=",minPts),
         xlab="x",ylab="y",asp=1)
    print(dbscan_result)
  }
}
```



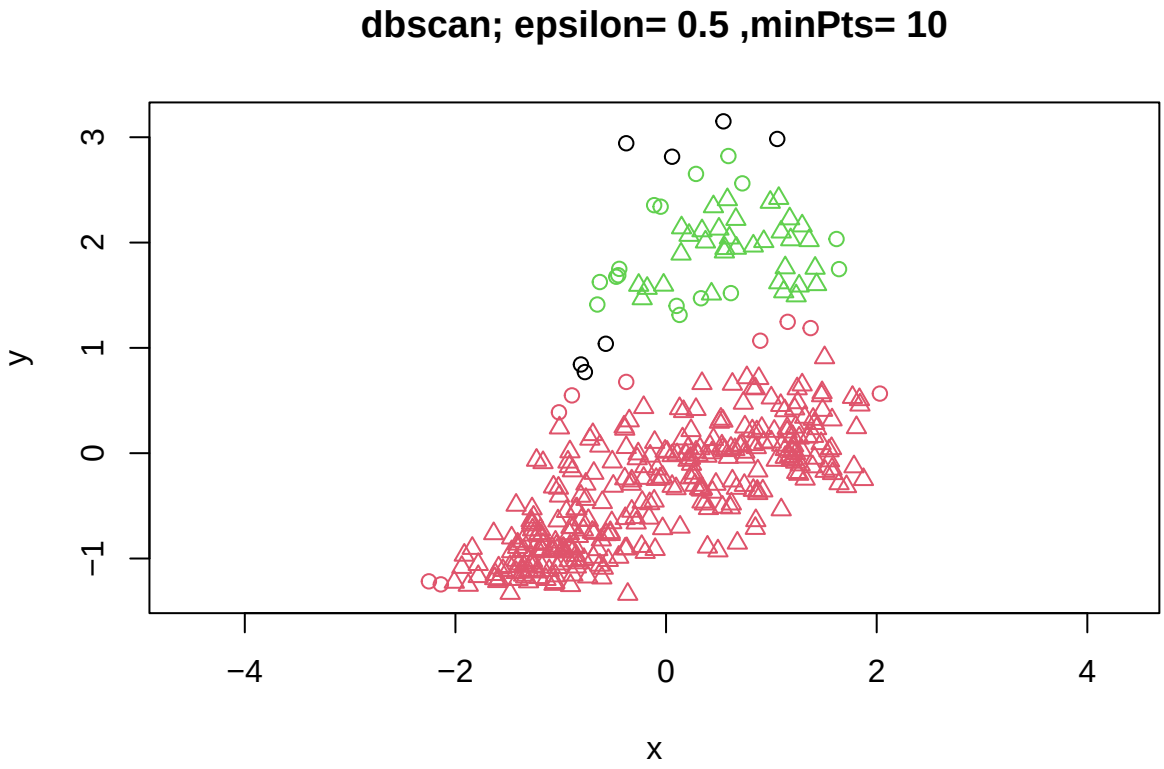

```
## dbscan Pts=366 MinPts=10 eps=0.25
##          0   1
## border 105  54
## seed     0 207
## total  105 261
```



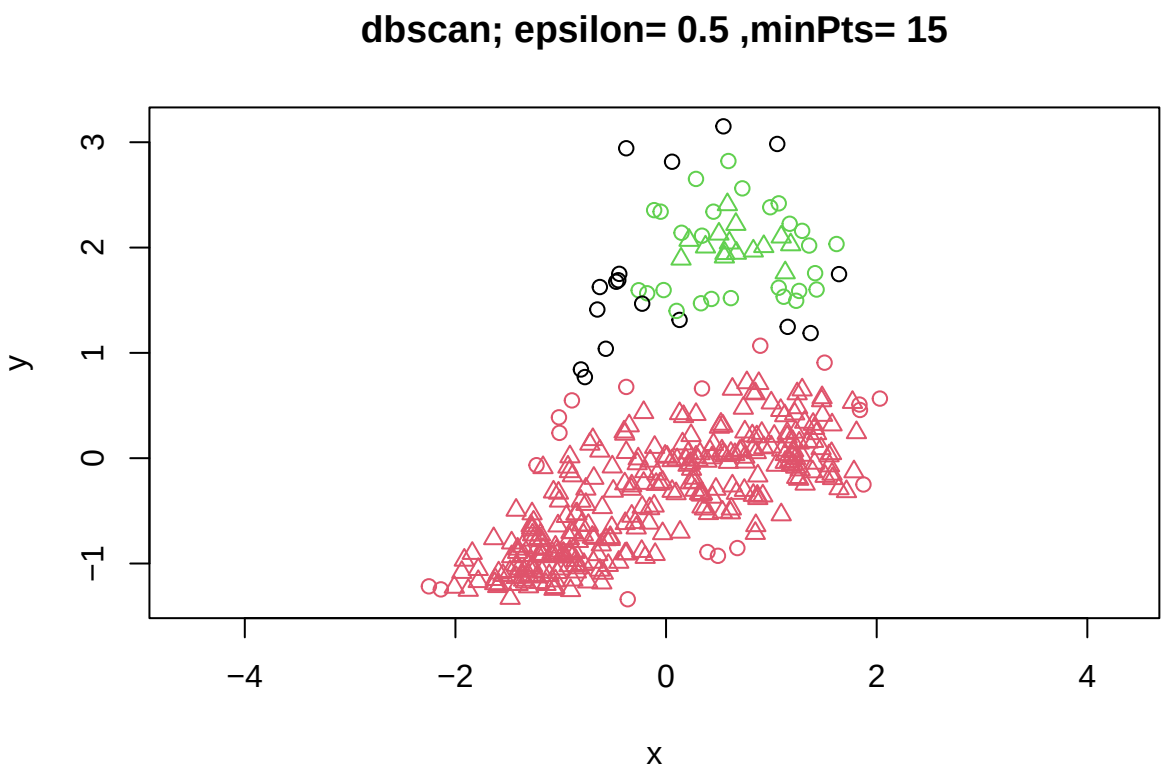

```
## dbscan Pts=366 MinPts=15 eps=0.25
##          0   1  2   3
## border 135  34 14  44
## seed     0  76  1  62
## total  135 110 15 106
```



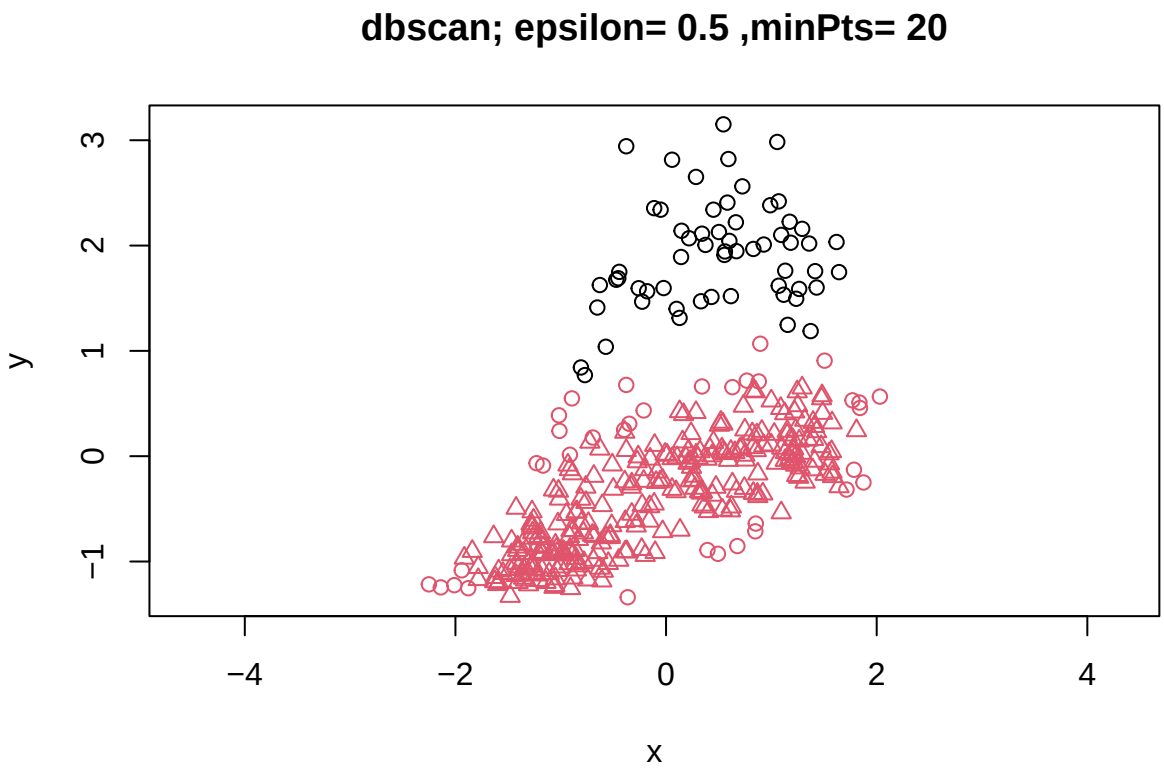

```
## dbscan Pts=366 MinPts=20 eps=0.25
##          0   1  2  3
## border 178  42 32 27
## seed     0  62  6 19
## total  178 104 38 46
```




```
## dbscan Pts=366 MinPts=10 eps=0.5
##        0   1  2
## border 7   9 16
## seed   0 300 34
## total  7 309 50
```



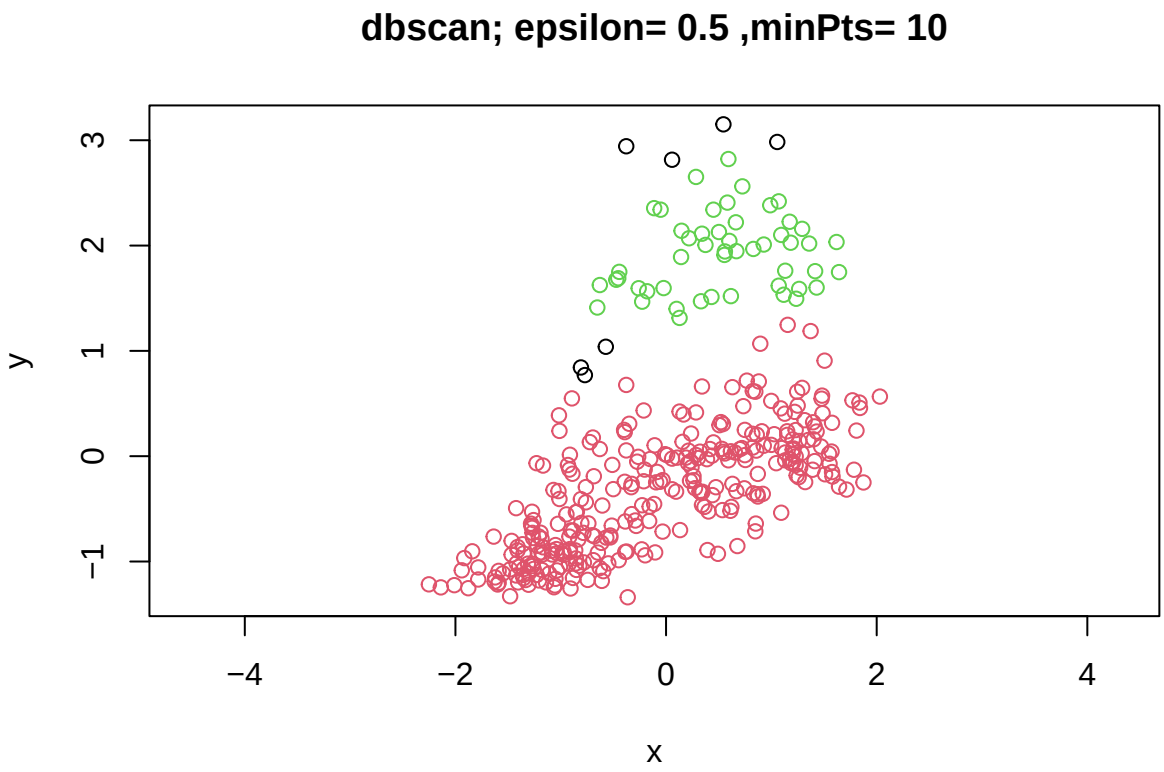

```
## dbscan Pts=366 MinPts=15 eps=0.5
##          0   1  2
## border 17  18 27
## seed    0 289 15
## total  17 307 42
```




```
## dbscan Pts=366 MinPts=20 eps=0.5
##         0   1
## border 59  35
## seed    0 272
## total  59 307
```

The resulting plots vary in the number of clusters and number of outliers, when trying different values for epsilon and for minimum points. Out of all, the combination of the tuning parameters that we consider the best one is the one having epsilon = 0.5 and minPts = 10. The reason why is that two clear clusters can be seen and out of the combinations is the one having the least amount of outliers, that is 7 outliers.

```
#Cross table
favorite_eps <- 0.5
favorite_minPts <- 10
dbscan_result <- dbscan::dbscan(Xs, eps = favorite_eps, minPts = favorite_minPts)

plot(dbscan_result, Xs, main=paste("dbscan; epsilon=",eps,",minPts=",favorite_minPts),
      xlab="x",ylab="y",asp=1)
```



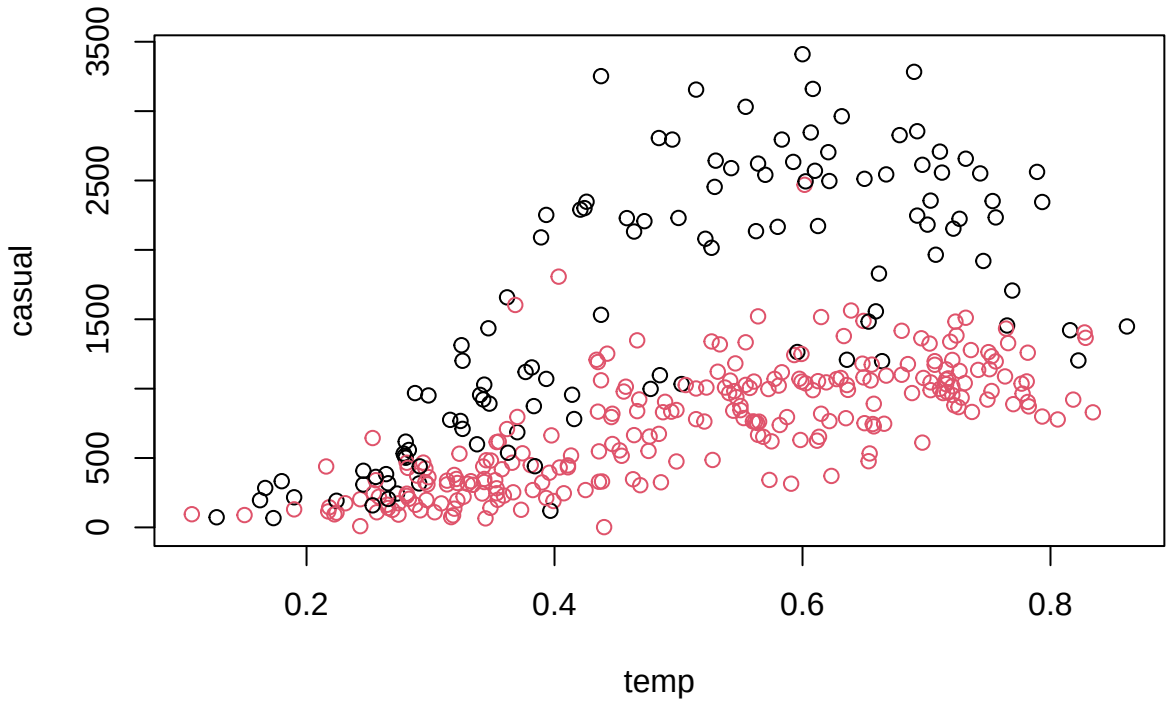

```
cluster_assignments <- dbscan_result$cluster

comparison_data <- data.frame(
  DBSCAN = cluster_assignments,
  mergenormals = cmnbhat$clustering
)

cross_table <- table(comparison_data)

print(cross_table)
```

```
##       mergenormals
## DBSCAN   1   2
##      0   0   7
##      1 306   3
##      2   0  50
```

Comparing the DBSCAN clustering with our favorite combination of tuning parameters with the results of mergenormals, we can see that the results are almost the same. There are only 3 points where the results from DBSCAN and mergenormals differ, because these 3 points correspong to cluster 1 for DBSCAN and to cluster 2 for mergenormals. This is the only difference, a part from the appearence of outliers.

**7.**

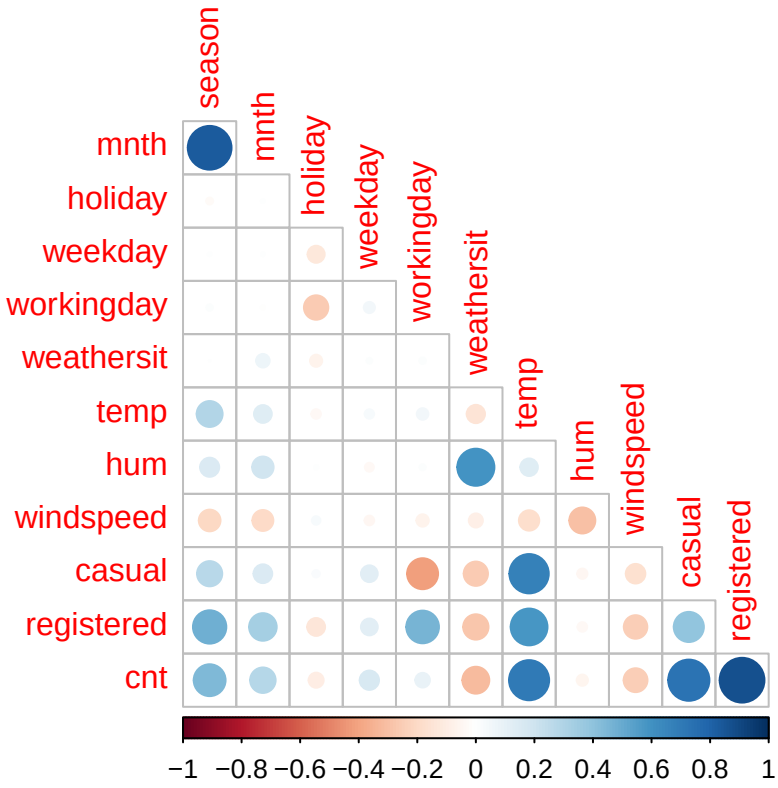
```
weekdays <- as.matrix(day[day$yr==1,c(8)])
plot(X, col=1+weekdays)
```




We plotted the data separated by weekday (red) and weekend (black). Here we can observe a similar separation as the plots with two clusters (mergenormals and DBSCAN). Hence it is apparent that the clusters are coming from the weekday/weekend separation. This is reasonable since we are looking at the data of casual users. Those are more likely to use the bikes on weekends than the registered users. So at the weekend when people are going on trips, the number of casual users increases.

But we can also see that the data is not perfectly separated. There are some points of the weekend in the weekday cluster. This happens mainly on the left side of the plot, at lower temperature. So at low temperature there are many points with a low number of casual users. Because the density is higher there, Mclust found a third cluster there. However, regarding the weekday/weekend separation, the separartion into two clusters is more reasonable.

```
M <- cor(as.matrix(day[day$yr==1,c(3,5,6,7,8,9,10,12,13,14,15,16)]), method="spearman")

corrplot(M, type="lower", diag = FALSE)
```




To reinforce our conclusions, we have also checked if there is another variable with a strong correlation (using spearman, cause we assume this variables are non-normally distributed) with **casual**. In the correlation plot above, we can see that the variables **workingday** and **temp** show the highest correlations with the **casual** variable. These are the other two variables we are considering for our final interpretation.