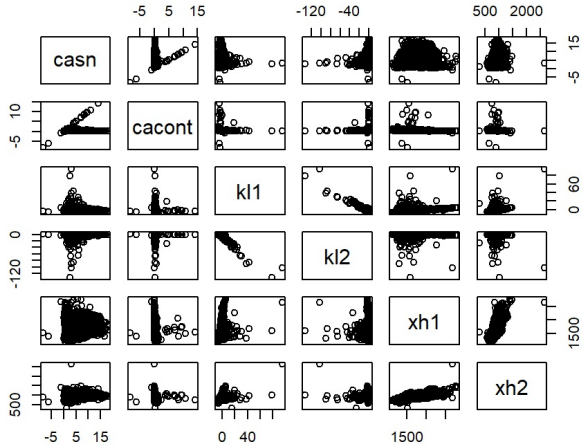
ex6_Palombarini

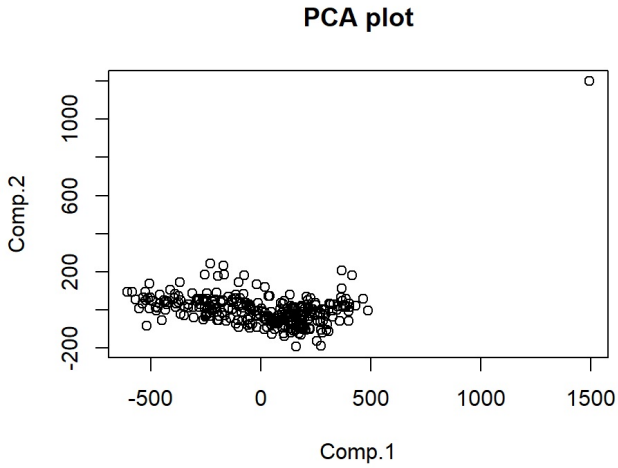
jacopo

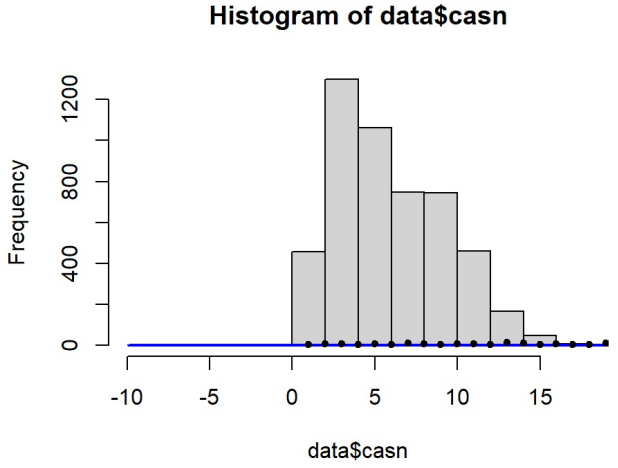
2023-11-24

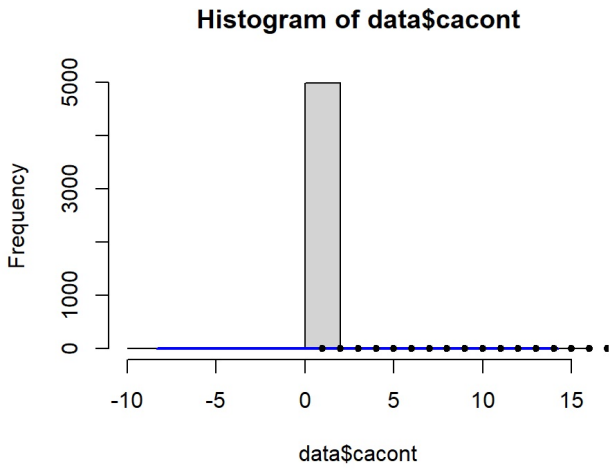
Exercice 1

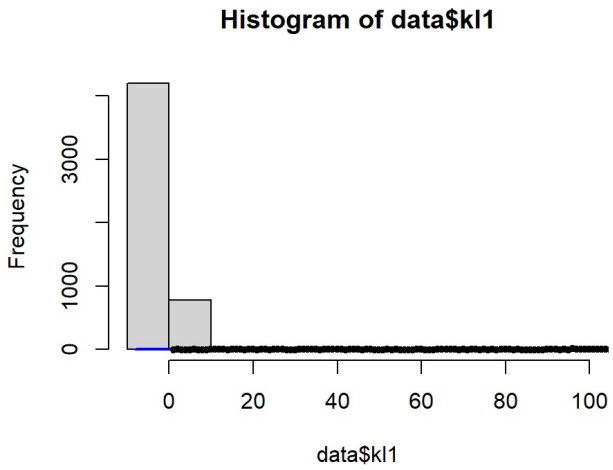


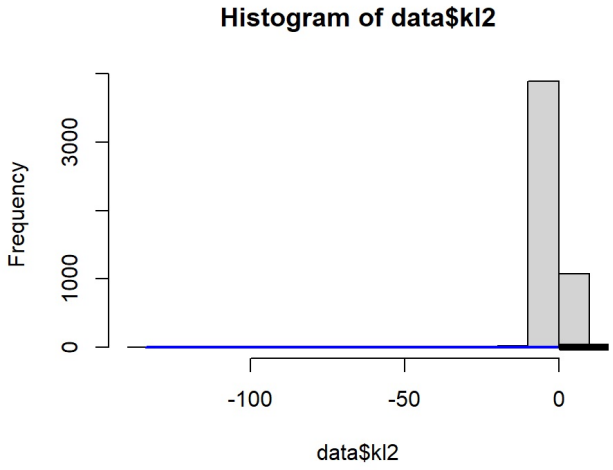
It seems that a Mixture model could be usefull, difficult to say which shape between gaussian and student is the more adapt.

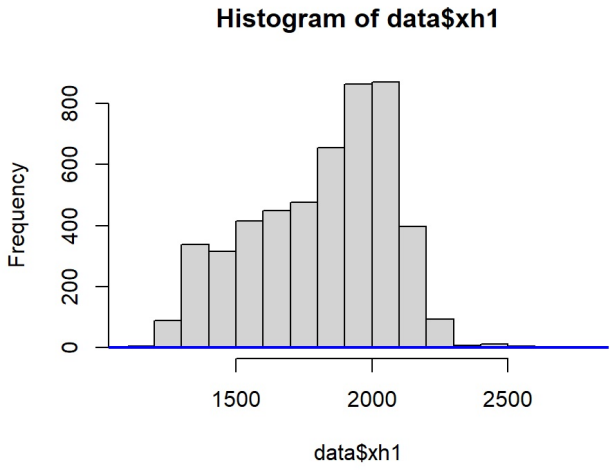


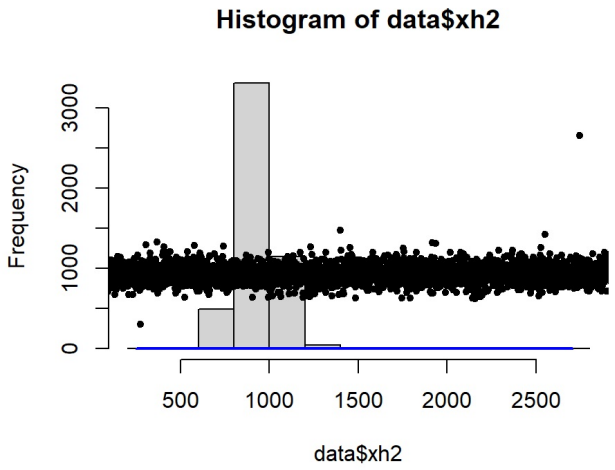


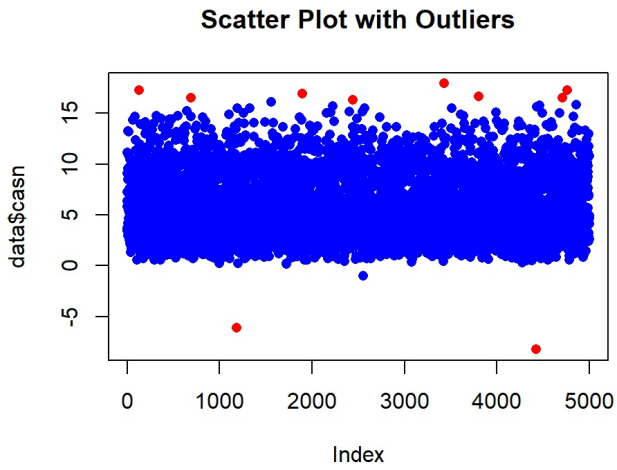

Histogram of data$cacont
Frequency
0
1000
3000
5000
-10
-5
0
5
10
15
data$cacont

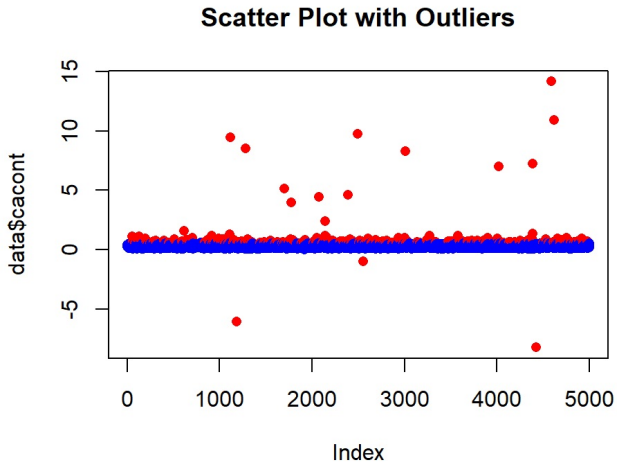

Histogram of data$kl1
Frequency
0
1000
3000
0
20
40
60
80
100
data$kl1

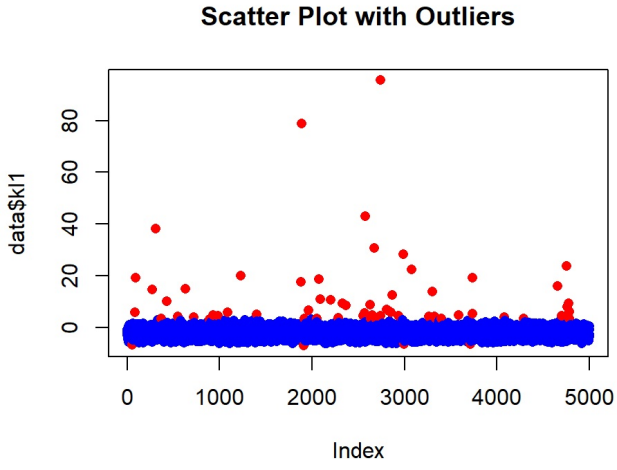

Histogram of data$kl2
Frequency
0
1000
3000
-100
-50
0
data$kl2

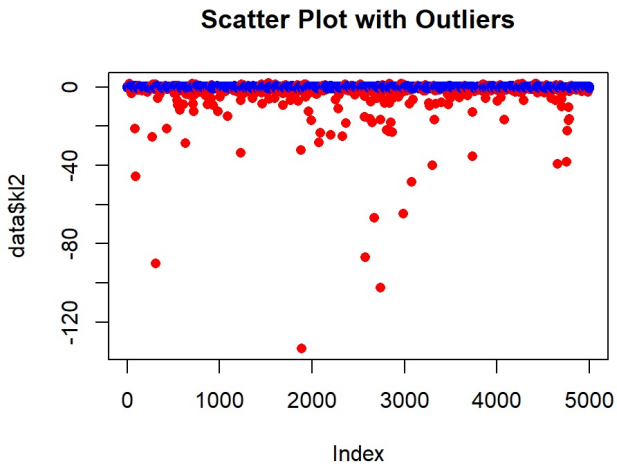

Histogram of data$xh1
Frequency
0
200
400
600
800
1500
2000
2500
data$xh1

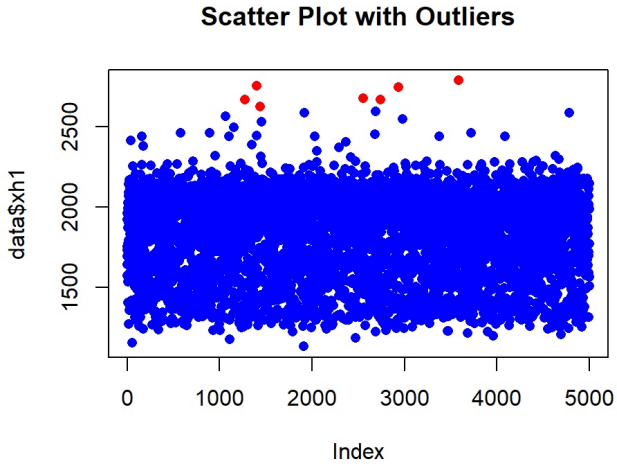

Histogram of data$xh2
Frequency
0
1000
2000
3000
500
1000
1500
2000
2500
data$xh2

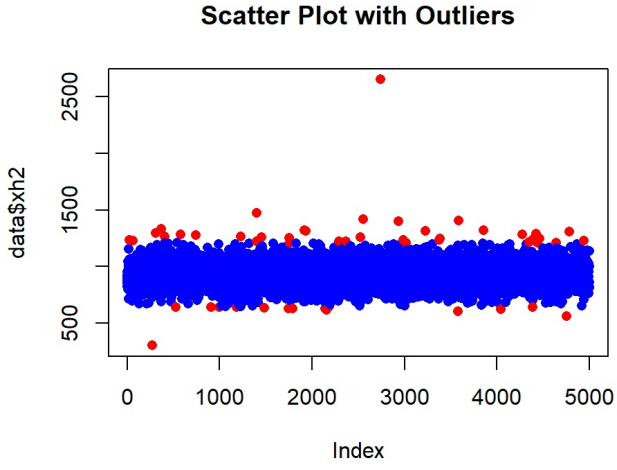

Scatter Plot with Outliers
data$casn
-5
0
5
10
15
0
1000
2000
3000
4000
5000
Index


Scatter Plot with Outliers
data$cacont
-5
0
5
10
15
0
1000
2000
3000
4000
5000
Index

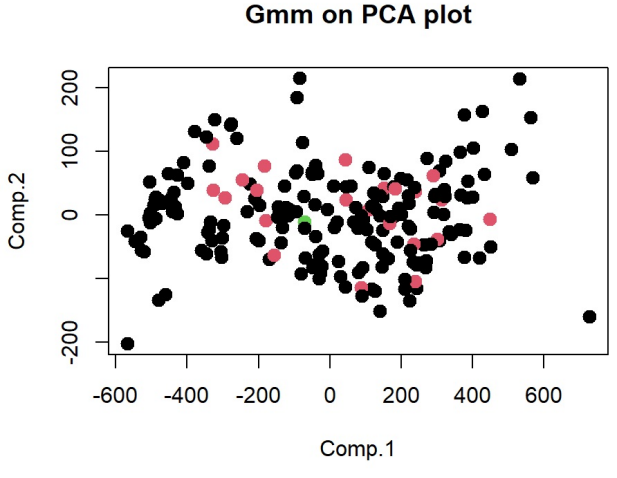

Scatter Plot with Outliers
data$kl1
0
20
40
60
80
0
1000
2000
3000
4000
5000
Index



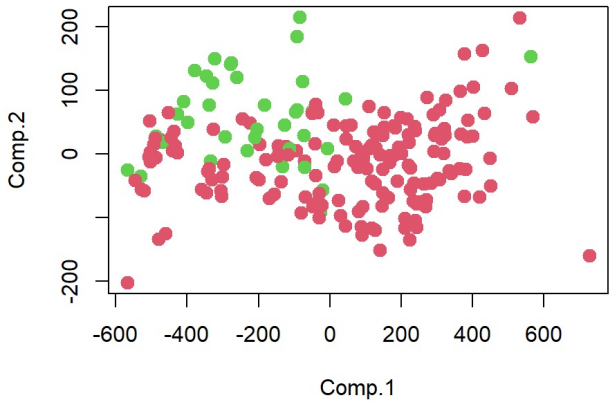




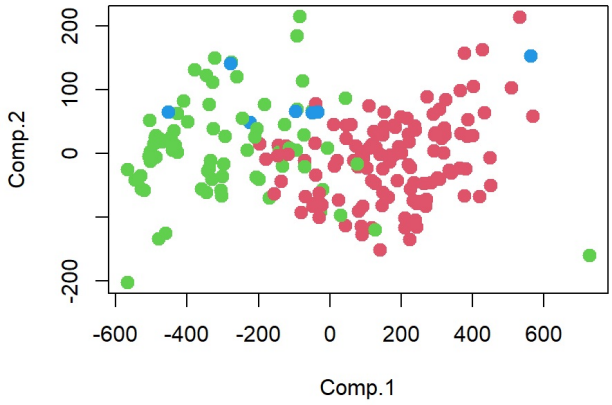
We use these plots to understand how variables are distributed, if data are spherical or not and if there is a consistent number of outliers (the answer is yes in this case).



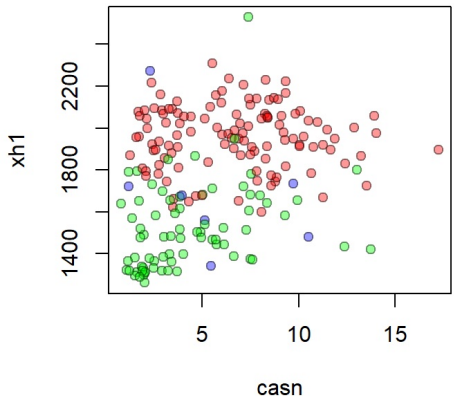
G.skewed on PCA plot



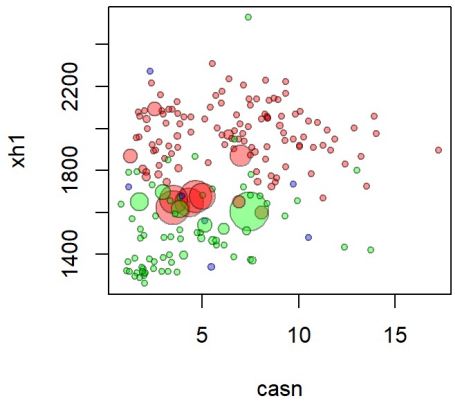
t-student on PCA plot

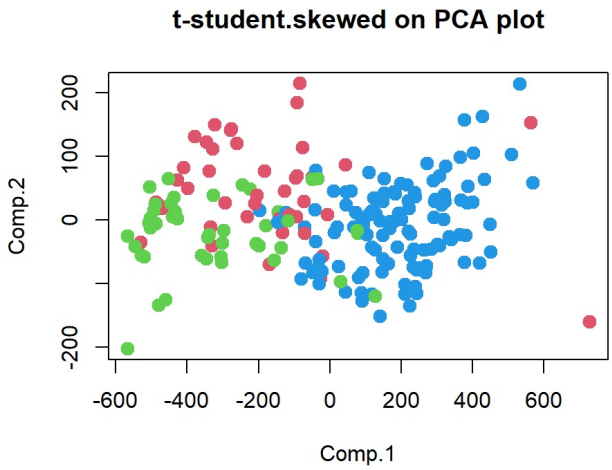


Marginal Contour Plot



Uncertainty Plot





Gaussian mixture seems the best one.

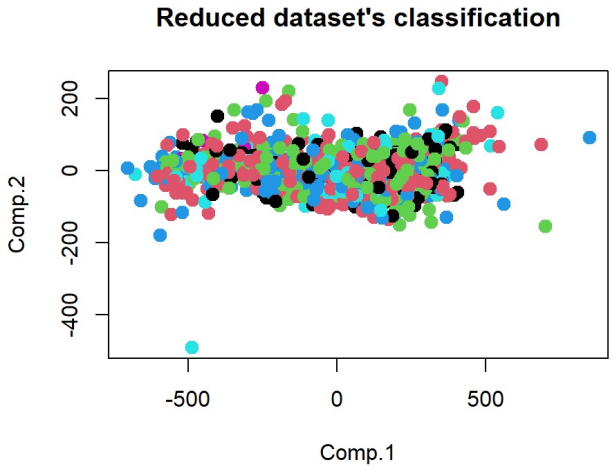
```
##      Standardized data        Original data
## [1,] "Gmm 4074.122"           "Gmm: 9036.456"
## [2,] "Skewed Normal: 4084.56" "Skewed Normal: 9008.31"
## [3,] "T-student: 4051.5"      "T-student: 8733.43"
## [4,] "T-skewed: Not found"    "T-skewed: 8700.64"
```

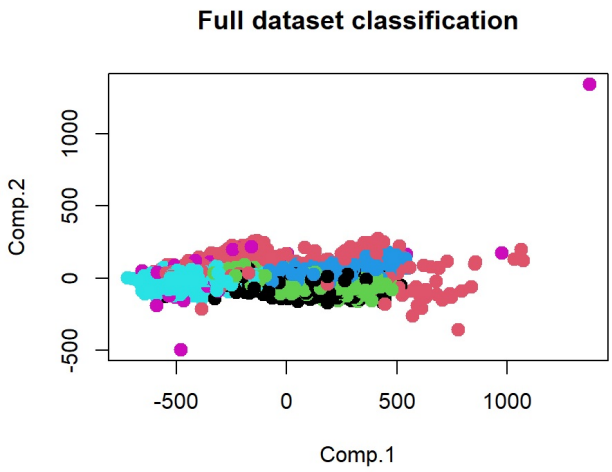
I had to reduce dimensionality and boundaries in the functions because in several cases they gave me problems and algorithms didn't converge. Cannot say which is the best model for standardized data but the pattern seems to respect the results for original data, so we can say that t.skewed is the best one (last function gave me problems and even with the lowest boundaries and a dataset of 300 observations I did'nt obtain a result form it). We also notice how performance heavily increases standardizing data.

Exercice 2

```
## [1] "We gained 8.48 seconds"
```

Seems pretty good.





Full dataset seems to be way better fitted than the reduced one, let's check with BIC measures

```
##      [,1]
## [1,] "Full dataset 141075.5"
## [2,] "Reduced dataset 29088.43"
```

It seems that they have proportionally similar results, but I think that comparing BIC of models that use different datasets could not be theoretically correct. So, relying on the plots of PCA, I'd say that even though we gained time, this method doesn't give us better results, in this specific case.

3

- Gaussian mixture model with fully flexible covariance matrices: mean: 10 variables × 4 components = 40 covariance matrix: 10×(10+1)/2 × 4 components = 220 mixing components: 4−1 (because they sum to 1) = 3 tot: 40+220+3=263.
- Gaussian mixture model with spherical covariance matrices: Component have a single variance parameter

mean: 40 parameters. variance: 10 variables x 4 components = 40 mixing components: 3 tot: 40+40+3=83

- Fully flexible skew-normal mixture: mean: 40 scale: 10×(10+1)/2 x 4 components = 220 shape: 4 components = 4 mixing components: 3 tot: 40+220+4+3=267
- Fully flexible mixture of multivariate t distributions: Here we also have df

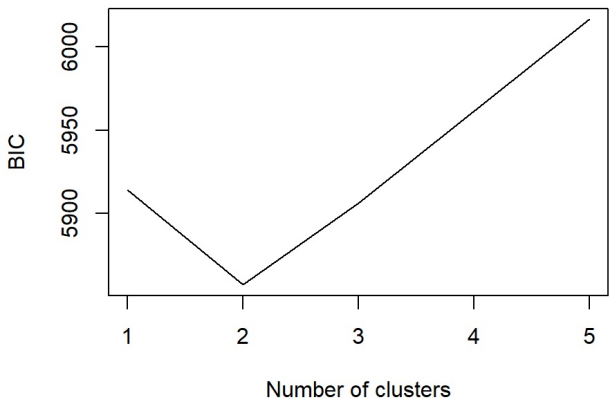
mean: 40 Scale: 10×(10+1)/2 x 4 components = 220 df: 4 components = 4 mixing components: 3 tot: 40+220+4+3=267

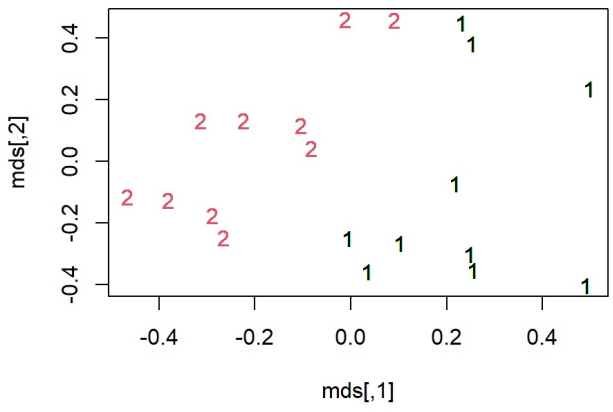
- Mixture of skew-t distributions with equal parameters: df and cov equals

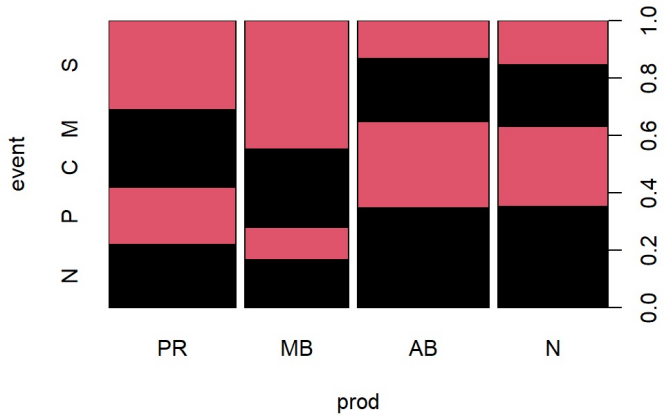
mean: 40 scale: 10×(10+1)/2=55 mixing components: 3 tot: 40+55+3=98

4

```
##     event
## prod  S  M  C  P  N
##   PR 84 35 38 54 59
##   MB 98 42 18 25 36
##   AB 37 32 29 84 96
##   N  36 28 22 65 82
```








```
##              estimated_clusters
## true_clusters   1   2
##             1 322  78
##             2 127 473
```

Estimation seems pretty good.

```
## [1] 0.3937807
```

Simple matching distance seems to perform worst. This is probably due to the fact that sm combined with average linkage are not that good at treating outliers, and as we've seen this variables are full of outliers.