Statistical Learning with High-Dimensional Data

DSTI A20 Cohort

Ariel Nataf

For some unknown reason to me the plot function doesn't show me the clusters colors. I have to knit the document to see the colored result it took me a lot of time to figure this way to avoid the problem

#Exercise 3 ## 3,1 Loading the Data

```
load("/Users/arielnataf/Desktop/DSTI/SLHD/Velib.Rdata")
```

3.2 Pretreatment et descriptive analysis

We look at the documentation first

```
?velibCount
```

```
## No documentation for 'velibCount' in specified packages and libraries:
## you could try '??velibCount'
```

We find very informative metadata: > The format is: > - data: the nb of available bikes of the 1189 stations at 181 time points. > - position: the longitude and latitude of the 1189 bike stations. > - dates: the download dates. > - bonus: indicates if the station is on a hill (bonus = 1). > - names: the names of the stations.

We also look for any missing data

```
is.na(Velib)
```

```
##     data position    dates    bonus    names
##    FALSE    FALSE    FALSE    FALSE    FALSE
```

There isn't any.

```
# We keep only the mean number of velib at each station
Velib_mean <- rowMeans(Velib$data)
data = cbind(Velib$position, Velib_mean)
data = cbind(Velib$bonus, data)

data
```

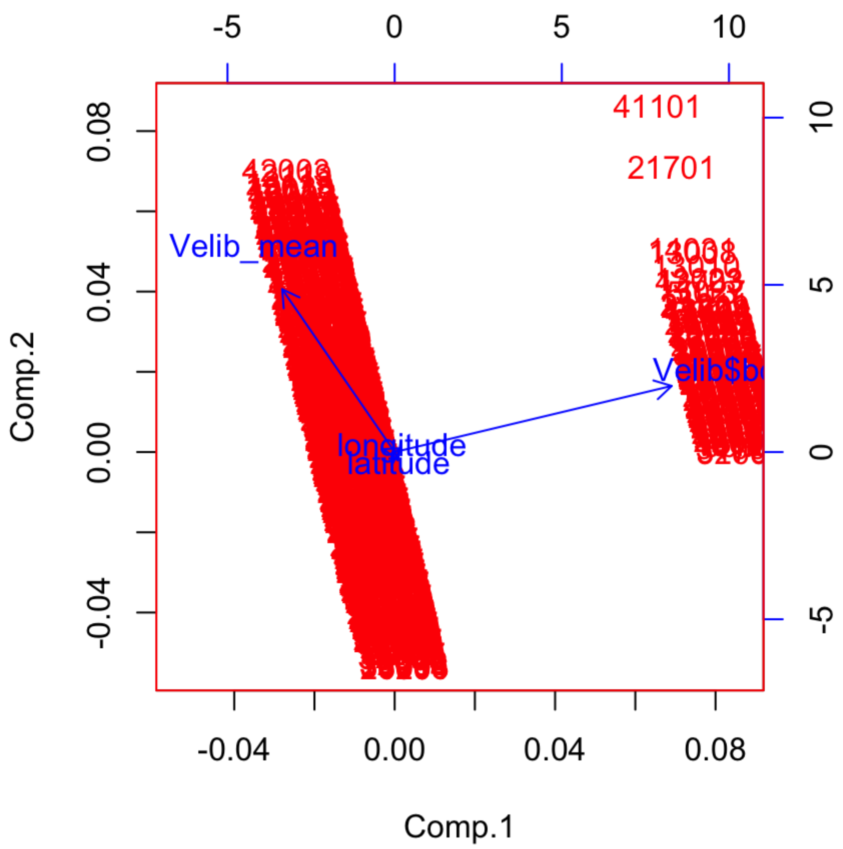
	Velib$bonus <dbl>	longitude <dbl>	latitude <dbl>	Velib_mean <dbl>
19117	0	2.377389	48.88630	0.27024503
17111	0	2.317591	48.89002	0.48307602
6103	0	2.330447	48.85030	0.44742171

	Velib$bonus <dbl>	longitude <dbl>	latitude <dbl>	Velib_mean <dbl>
15042	0	2.271396	48.83373	0.46416320
12003	0	2.366897	48.84589	0.56110053
13038	0	2.363335	48.82191	0.33473149
17041	0	2.287667	48.88288	0.39776023
41203	0	2.455529	48.85013	0.38078215
43401	0	2.464026	48.81995	0.64223450
5015	0	2.349983	48.84151	0.47270809

1-10 of 1,189 rows Previous **1** 2 3 4 5 6 ... 119 Next

3.3 Data visualization

```
X = data
?princomp
# The usual manner to do PCA in R
pca = princomp(X)
biplot(pca,col=c("red","blue"))
```




```
?biplot
```

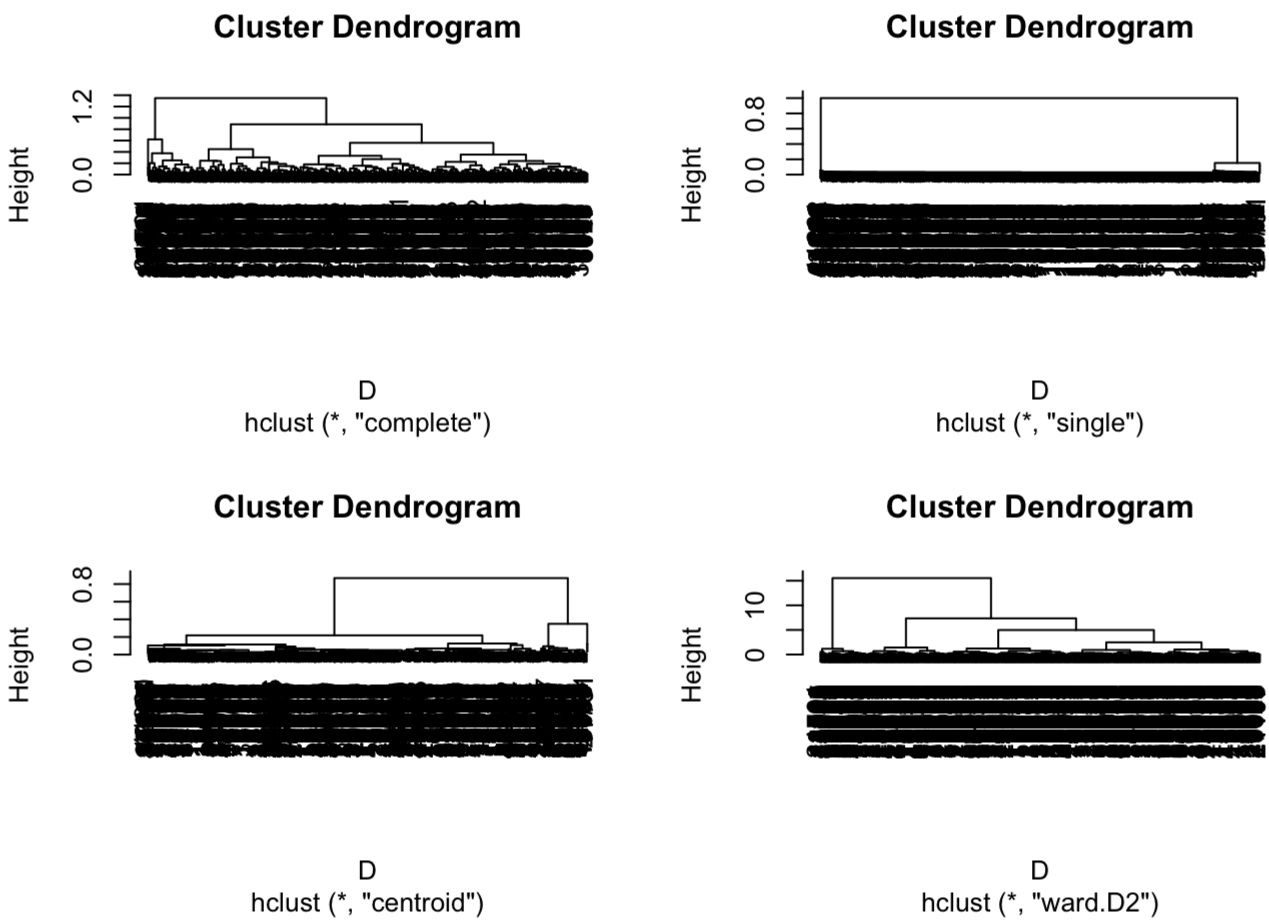
Looking at the PCA, • we can see that longitude and latitude arrows stay in the middle little influence • Velib$bonus (on a hill) goes to the right • The mean number of available velibs goes to the left

Velib$bonus and velib at date are opposite. We can guess a hill has an impact on the usage of a velib

3.4 Clustering

3.4.1 Hierarchical clustering

```
D = dist(data)
par(mfrow=c(2,2))
hc1 = hclust(D,method = "complete"); plot(hc1)
hc2 = hclust(D,method = "single"); plot(hc2)
hc3 = hclust(D,method = "centroid"); plot(hc3)
hc4 = hclust(D,method = "ward.D2"); plot(hc4)
```

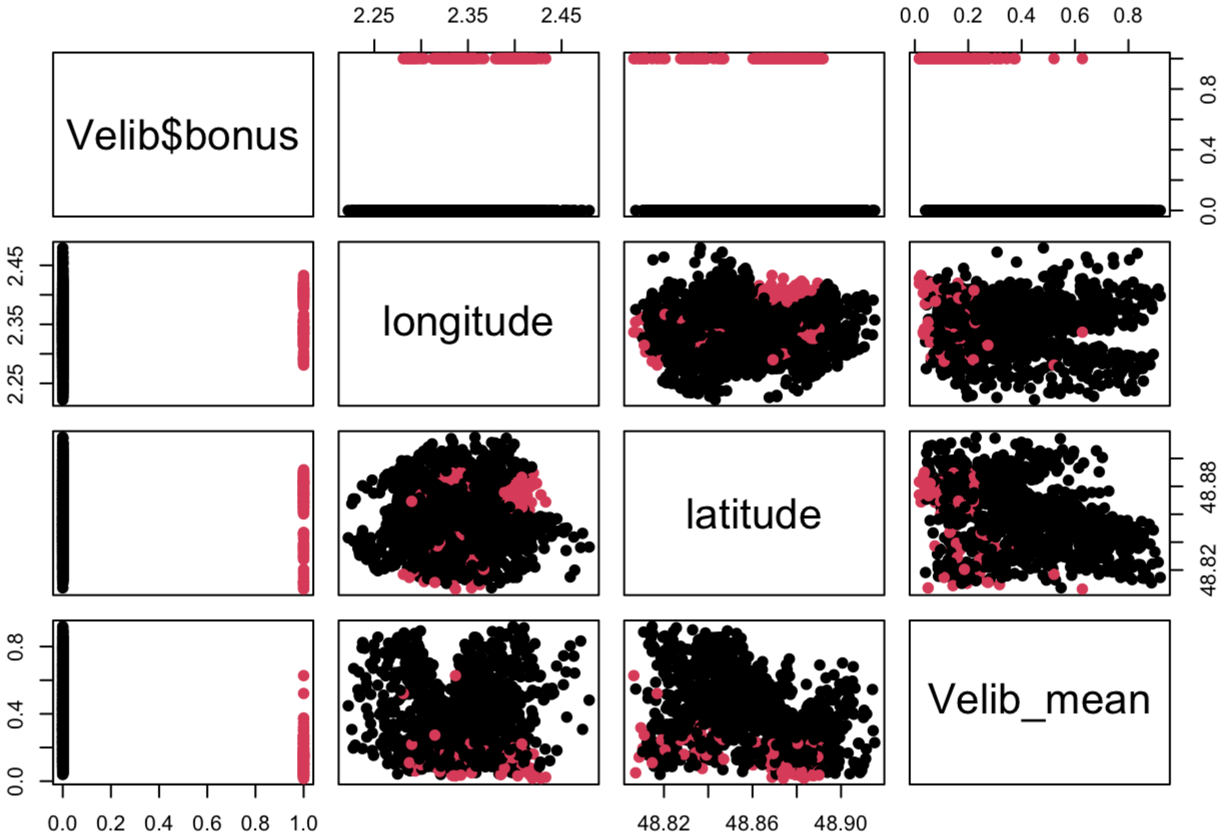


Complete seems to be the best method looking at the dendograms

```
hc1 = hclust(D,method = "complete")
cl1 = cutree(hc4,k = 2)
cl1
```

We looks at clusters with all variables

```
plot(data,col=cl1,pch=19)
```



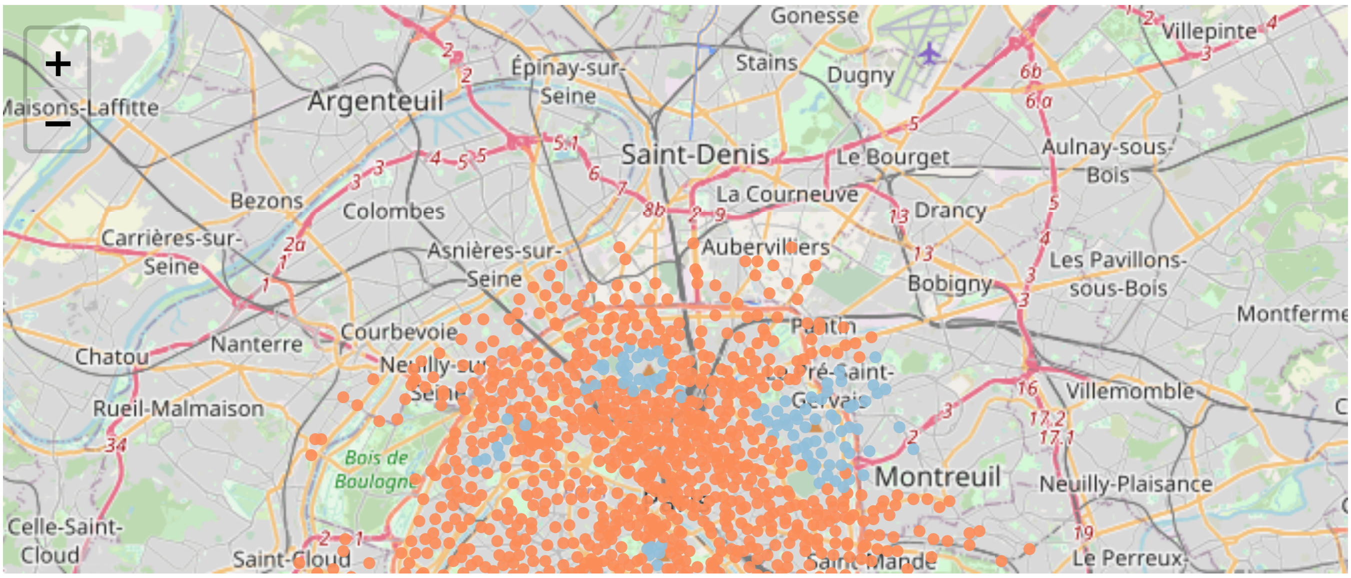
clusters depending of hill and the mean numbers are clearly there

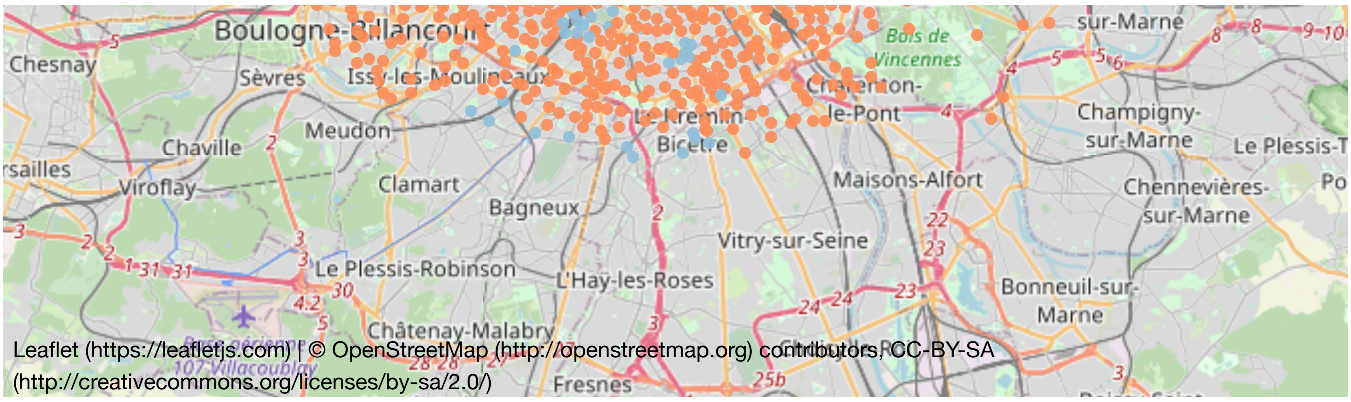
Looking at the clusters on a pretty map:

```
#install.packages("leaflet")
library(leaflet)

palette = colorFactor("RdYlBu", domain = NULL)
leaflet(X) %>% addTiles() %>%
    addCircleMarkers(radius = 3,
                     color = palette(cl1),
                     stroke = FALSE, fillOpacity = 0.9)
```

```
## Assuming "longitude" and "latitude" are longitude and latitude, respectively
```





very pretty, I can see my home

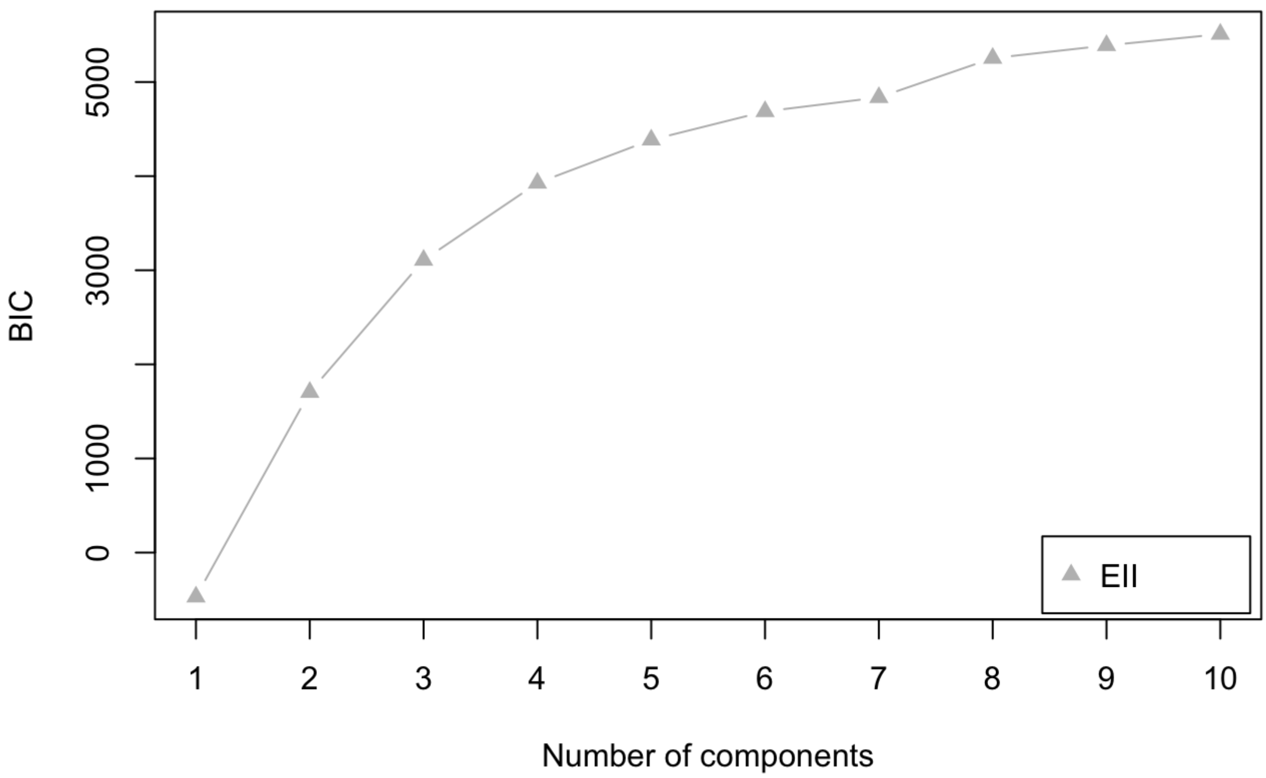
On the map we can guess the second cluster is the hills.

3.4.2 k-means

```
#install.packages("mclust")
library(mclust)
```

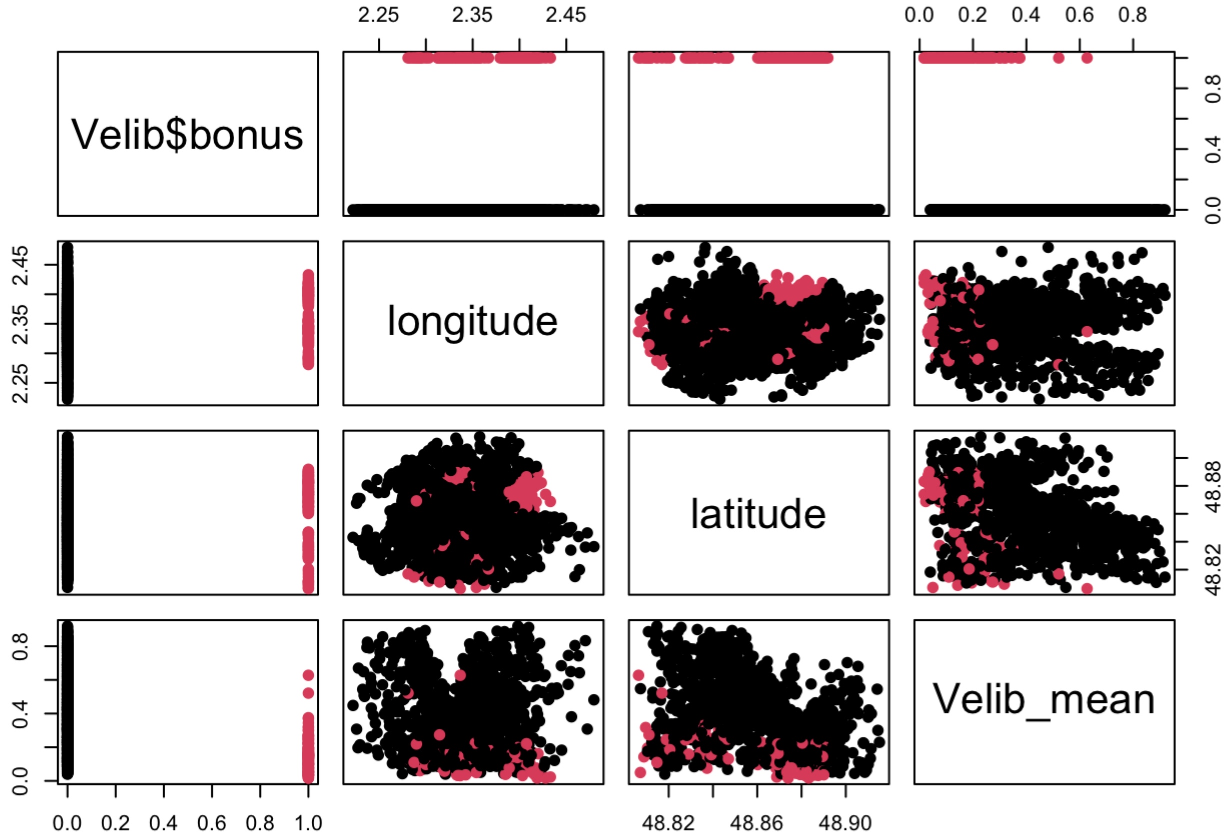
```
## Package 'mclust' version 5.4.7
## Type 'citation("mclust")' for citing this R package in publications.
```

```
out = Mclust(data[c("Velib_mean", "Velib$bonus")],G=1:10,modelNames = "EII")
plot(out, what = 'BIC')
```

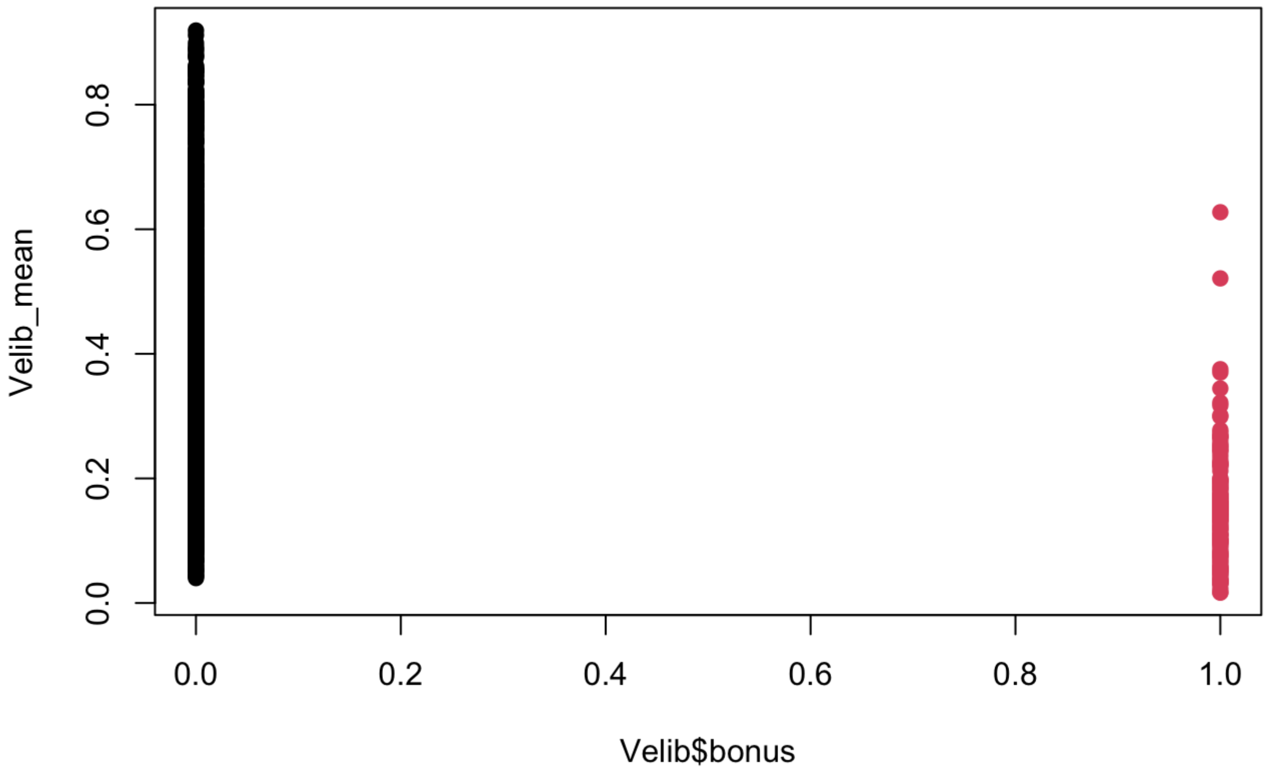


We only use 2 clusters

```
out2 = kmeans(data, centers = 2, nstart = 10)
plot(data,col=out2$cluster,pch=19)
```

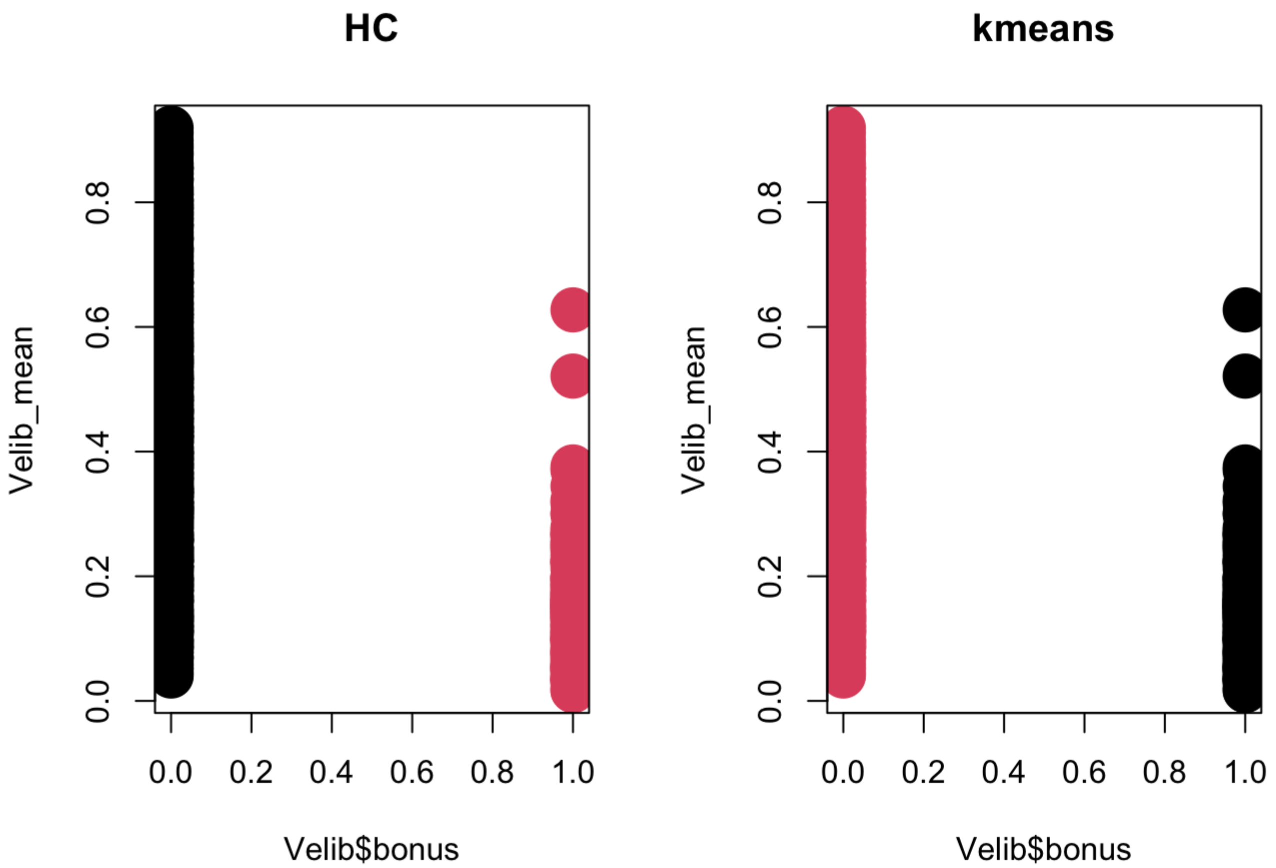



```
plot(data[,c(1,4)],col=out2$cluster,pch=19)
```



Comparison with kmeans and hclust on mean number of velib on a station and hills

```
par(mfrow = c(1,2))
D = dist(data)
hc = hclust(D,method = "complete"); out.hc = cutree(hc,2)
plot(data[,c(1,4)],col=out.hc,pch=19,cex = 3,main='HC')
out.km = kmeans(data, centers = 2, nstart = 10)
plot(data[,c(1,4)],col=out.km$cluster,pch=19,cex = 3,main='kmeans')
```



it

looks similar

3.5 Summary

People don’t use as much velibs on top of a hill than down. Otherwise west/north/west/east don’t impact very much.

Exercise 1 Ariel Nataf

1. To select the best number of cluster we can not use the maximum Likelihood (risk of overfitting)
We have to use the Bayesian Information Criterion (BIC) corresponding to the log likelihood with a penalty.

We can use R to compare the BIC scores of different models (different number of cluster).

2. Double Cross Validation is used to compare the results of different models, looking at the performances (error) and in case of K Nearest Neighbours, even help select the best K.

For the set up we pick 2 models. Usually LDA is the reference because it gives good results in general and is relatively simple.
We train the data with each model on a sample and compute the errors with a test sample (with different observations than the training sample).

Exercise 2
1. Within a cluster lower the variance, more compact the cluster is.

2.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
Var 1	0	0	1	3	3.5	1	3	4
Var 2	1	2	1	1	1	5	4	5

assignment to the 2 groups of closest center $var_1 \{x_1, x_2, x_6, x_7, x_8\}$
$var_2 \{x_4, x_5\}$
we substract the mean to the inial centres, we get new values for the var, we repeat until the clusters are fixed.