Data Analysis Project - Velib in Paris

Authors: Yasmine BOULKAID, Edda IVELAND, Vilde OPDAL, Laura VAZQUEZ

Institut National des Sciences Appliquées de Toulouse, 2023-2024
4th year, Génie de Mathématiques et Modélisation

Part 1: An introduction to the data (descriptive statistics)

1.1. Uploading and verification of the data

In [1]:
```
install.packages('ggplot2')
install.packages('reshape2')
install.packages('gridExtra')
install.packages("mclust")
install.packages("cluster")
install.packages("factoextra")
install.packages("FactoMineR")
install.packages("ppclust")
install.packages("reticulate")
install.packages("reshape")
install.packages("corrplot")
install.packages("circlize")
install.packages("tidyverse")
install.packages("ggpubr")
install.packages("ggmap")
install.packages("cvms")
install.packages("ggimage")
install.packages("rsvg")
install.packages('Rfast')
```

```
Les packages binaires téléchargés sont dans
        /var/folders/d_/vz74p2wd2130_h4z5g1ch3vc0000gn/T//Rtmpbjnj4k/downloa
ded_packages

Les packages binaires téléchargés sont dans
        /var/folders/d_/vz74p2wd2130_h4z5g1ch3vc0000gn/T//Rtmpbjnj4k/downloa
ded_packages

Les packages binaires téléchargés sont dans
        /var/folders/d_/vz74p2wd2130_h4z5g1ch3vc0000gn/T//Rtmpbjnj4k/downloa
ded_packages

Les packages binaires téléchargés sont dans
        /var/folders/d_/vz74p2wd2130_h4z5g1ch3vc0000gn/T//Rtmpbjnj4k/downloa
ded_packages

Les packages binaires téléchargés sont dans
        /var/folders/d_/vz74p2wd2130_h4z5g1ch3vc0000gn/T//Rtmpbjnj4k/downloa
ded_packages

Les packages binaires téléchargés sont dans
        /var/folders/d_/vz74p2wd2130_h4z5g1ch3vc0000gn/T//Rtmpbjnj4k/downloa
ded_packages

Les packages binaires téléchargés sont dans
        /var/folders/d_/vz74p2wd2130_h4z5g1ch3vc0000gn/T//Rtmpbjnj4k/downloa
ded_packages

Les packages binaires téléchargés sont dans
        /var/folders/d_/vz74p2wd2130_h4z5g1ch3vc0000gn/T//Rtmpbjnj4k/downloa
ded_packages

Les packages binaires téléchargés sont dans
        /var/folders/d_/vz74p2wd2130_h4z5g1ch3vc0000gn/T//Rtmpbjnj4k/downloa
ded_packages

Les packages binaires téléchargés sont dans
        /var/folders/d_/vz74p2wd2130_h4z5g1ch3vc0000gn/T//Rtmpbjnj4k/downloa
ded_packages

Les packages binaires téléchargés sont dans
        /var/folders/d_/vz74p2wd2130_h4z5g1ch3vc0000gn/T//Rtmpbjnj4k/downloa
ded_packages

Les packages binaires téléchargés sont dans
        /var/folders/d_/vz74p2wd2130_h4z5g1ch3vc0000gn/T//Rtmpbjnj4k/downloa
ded_packages

Les packages binaires téléchargés sont dans
        /var/folders/d_/vz74p2wd2130_h4z5g1ch3vc0000gn/T//Rtmpbjnj4k/downloa
ded_packages

Les packages binaires téléchargés sont dans
        /var/folders/d_/vz74p2wd2130_h4z5g1ch3vc0000gn/T//Rtmpbjnj4k/downloa
ded_packages
```

```
Les packages binaires téléchargés sont dans
	/var/folders/d_/vz74p2wd2130_h4z5g1ch3vc0000gn/T//Rtmpbjnj4k/downloa
ded_packages

Les packages binaires téléchargés sont dans
	/var/folders/d_/vz74p2wd2130_h4z5g1ch3vc0000gn/T//Rtmpbjnj4k/downloa
ded_packages

Les packages binaires téléchargés sont dans
	/var/folders/d_/vz74p2wd2130_h4z5g1ch3vc0000gn/T//Rtmpbjnj4k/downloa
ded_packages

Les packages binaires téléchargés sont dans
	/var/folders/d_/vz74p2wd2130_h4z5g1ch3vc0000gn/T//Rtmpbjnj4k/downloa
ded_packages

Les packages binaires téléchargés sont dans
	/var/folders/d_/vz74p2wd2130_h4z5g1ch3vc0000gn/T//Rtmpbjnj4k/downloa
ded_packages
```

In [2]:

```
library(ggplot2)
library(reshape2)
library(gridExtra)
library(mclust)
library(cluster)
library(factoextra)
library(FactoMineR)
library(ppclust)
library(reticulate)
library(reshape)
library(corrplot)
library(circlize)
library(tidyverse)
library(ggpubr)
library(cluster)
library(ggmap)
register_stadiamaps("2f2ea565-d310-4610-9cae-b8e7dacae25d", write = TRUE)
library(cvms)
library(ggimage)
library(rsvg)
library(Rfast)
```

```
Warning message:
“le package ‘ggplot2’ a été compilé avec la version R 4.2.3”
Warning message:
“le package ‘mclust’ a été compilé avec la version R 4.2.3”
Package 'mclust' version 6.1.1
Type 'citation("mclust")' for citing this R package in publications.

Warning message:
“le package ‘cluster’ a été compilé avec la version R 4.2.3”
Welcome! Want to learn more? See two factoextra-related books at https://go
o.gl/ve3WBa

Warning message:
“le package ‘FactoMineR’ a été compilé avec la version R 4.2.3”
Warning message:
“le package ‘ppclust’ a été compilé avec la version R 4.2.3”
Warning message:
“le package ‘reticulate’ a été compilé avec la version R 4.2.3”

Attachement du package : ‘reshape’


Les objets suivants sont masqués depuis ‘package:reshape2’:

    colsplit, melt, recast


corrplot 0.92 loaded

Warning message:
“le package ‘circlize’ a été compilé avec la version R 4.2.3”
========================================
circlize version 0.4.16
CRAN page: https://cran.r-project.org/package=circlize
Github page: https://github.com/jokergoo/circlize
Documentation: https://jokergoo.github.io/circlize_book/book/

If you use it in published research, please cite:
Gu, Z. circlize implements and enhances circular visualization
  in R. Bioinformatics 2014.

This message can be suppressed by:
  suppressPackageStartupMessages(library(circlize))
========================================


Warning message:
“le package ‘readr’ a été compilé avec la version R 4.2.3”
Warning message:
“le package ‘dplyr’ a été compilé avec la version R 4.2.3”
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.
0 ──
✔ dplyr     1.1.4     ✔ readr     2.1.5
✔ forcats   1.0.0     ✔ stringr   1.5.0
✔ lubridate 1.9.3     ✔ tibble    3.2.1
✔ purrr     1.0.1     ✔ tidyr     1.3.0
```

```
── Conflicts ─────────────────────────────────────── tidyverse_conflicts
() ──
✖ dplyr::combine()   masks gridExtra::combine()
✖ tidyr::expand()    masks reshape::expand()
✖ dplyr::filter()    masks stats::filter()
✖ dplyr::lag()       masks stats::lag()
✖ purrr::map()       masks mclust::map()
✖ dplyr::rename()    masks reshape::rename()
✖ lubridate::stamp() masks reshape::stamp()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all c
onflicts to become errors
Warning message:
“le package ‘ggmap’ a été compilé avec la version R 4.2.3”
ℹ Google's Terms of Service: <https://mapsplatform.google.com>
  Stadia Maps' Terms of Service: <https://stadiamaps.com/terms-of-service/>
  OpenStreetMap's Tile Usage Policy: <https://operations.osmfoundation.org/p
olicies/tiles/>
ℹ Please cite ggmap if you use it! Use `citation("ggmap")` for details.
ℹ Replacing old key (2f2ea565) with new key in /Users/boulkaid/.Renviron

Warning message:
“le package ‘cvms’ a été compilé avec la version R 4.2.3”

Attachement du package : ‘cvms’


L'objet suivant est masqué depuis ‘package:ggpubr’:

    font



Attachement du package : ‘ggimage’


L'objet suivant est masqué depuis ‘package:ggmap’:

    theme_nothing


L'objet suivant est masqué depuis ‘package:ggpubr’:

    theme_transparent


Linking to librsvg 2.56.3

Warning message:
“le package ‘Rfast’ a été compilé avec la version R 4.2.3”
Le chargement a nécessité le package : Rcpp

Le chargement a nécessité le package : RcppZiggurat

Le chargement a nécessité le package : RcppParallel
```

```
Attachement du package : ‘RcppParallel’


L'objet suivant est masqué depuis ‘package:Rcpp’:

    LdFlags



Rfast: 2.1.0

 ___ __ __ __ __    __ __ __ __ __ _            _               __ __ __ __
__     __ __ __ __ __ __
|  __ __ __ __  |  |  __ __ __ __ _/           / \             |  __ __ __
__ /   /__ __ _   _ __ __\
| |           | |  | |                        / _ \            | |
/ /
| |           | |  | |                       / / \ \           | |
/ /
| |           | |  | |                      / /   \ \          | |
/ /
| |__ __ __ __| |  | |__ __ __ __          / /     \ \         | |__ __ __
__ _        / /__/\
|    __ __ __ __|  |  __ __ __ __|        / /__ _ __\ \        |_ __ __ __
_   |      / ___  /
|   \             | |                    / _ _ _ _ _ _ \
| |      \/  / /
| |\ \             | |                  / /          \ \
| |         / /
| | \ \             | |                / /            \ \
| |        / /
| |  \ \            | |               / /              \ \
| |       / /
| |   \ \__ __ _    | |              / /                \ \     _ __ __ __
_| |     / /
|_|    \__ __ __\  |_|             /_/                  \_\   /_ __ __ __
___|      \/            team


Attachement du package : ‘Rfast’


L'objet suivant est masqué depuis ‘package:dplyr’:

    nth


Les objets suivants sont masqués depuis ‘package:purrr’:

    is_integer, transpose


L'objet suivant est masqué depuis ‘package:mclust’:

    dmvnorm
```

In [3]:
```
load("velib.RData")
summary(velib)
```

```
         Length Class      Mode
data      181   data.frame list
position    2   data.frame list
dates     181   -none-     character
bonus    1189   -none-     numeric
names    1189   -none-     character
```

In [4]:
```
# data preparation
loading = as.matrix(velib$data)
colnames(loading) = 1:ncol(loading)
rownames(loading) = velib$names

stations = 1:nrow(loading)
coord = velib$position[stations,]
coord$bonus = velib$bonus[stations]

# select exactly 7 days of data (we remove the first 13 dates)
dates = 14:181
loading = loading[stations, dates]
colnames(loading) = 1:length(dates)

loading_hill = cbind(loading, coord$bonus)
colnames(loading_hill)[ncol(loading_hill)] = 'hill'


head(loading)
head(coord)
```

	1	2	3	4	5	6
EURYALE DEHAYNIN	0.03846154	0.03846154	0.07692308	0.03846154	0.03846154	0.03846154
LEMERCIER	0.47826087	0.47826087	0.47826087	0.43478261	0.43478261	0.43478261
MEZIERES RENNES	0.21818182	0.14545455	0.12727273	0.10909091	0.10909091	0.10909091
FARMAN	0.95238095	0.95238095	0.95238095	0.95238095	0.95238095	0.95238095
QUAI DE LA RAPEE	0.92753623	0.81159420	0.73913043	0.72463768	0.72463768	0.72463768
CHOISY POINT D'IVRY	0.16666667	0.16666667	0.16666667	0.16666667	0.16666667	0.16666667

A data.frame: 6 × 3

	longitude	latitude	bonus
	<dbl>	<dbl>	<dbl>
19117	2.377389	48.88630	0
17111	2.317591	48.89002	0
6103	2.330447	48.85030	0
15042	2.271396	48.83373	0
12003	2.366897	48.84589	0
13038	2.363335	48.82191	0

In [5]:
```
cat(' shape of loading dataframe', dim(loading), '\n', 'shape of coord dataf
```

```
shape of loading dataframe 1189 168
shape of coord dataframe 1189 3
```

In [6]:
```
# checking for missing values
cat('There are', sum(is.na(loading)), 'missing values in the loading data fr
```

```
There are 0 missing values in the loading data frame and 0 missing values in
the coord dataframe
```

In [7]:
```
# checking for duplicates in the data frame
cat('There are', sum(duplicated(loading)), 'duplicates in the loading data f
```

```
There are 0 duplicates in the loading data frame and 0 duplicates in the coo
rd dataframe
```

In [8]:
```
print('--- Average fill rate ---')
print(mean(loading))

print('--- Least crowded station, on average ---')
i = which.min(rowMeans(loading))
print(i)
print(rowMeans(loading)[i])

print('--- Fullest station, on average ---')
i = which.max(rowMeans(loading))
print(i)
print(rowMeans(loading)[i])
```

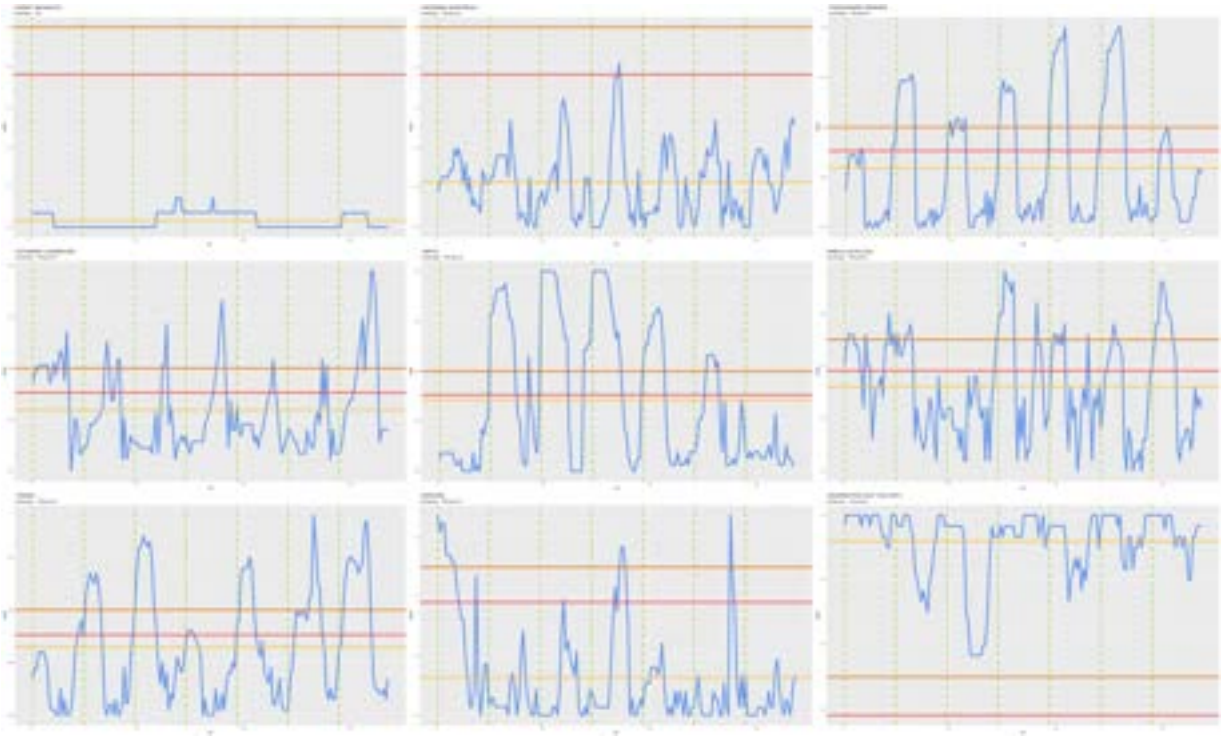
```
[1] "--- Average fill rate ---"
[1] 0.3816218
[1] "--- Least crowded station, on average ---"
 HORNET (BAGNOLET)
                997
 HORNET (BAGNOLET)
       0.01613284
[1] "--- Fullest station, on average ---"
 INSURRECTION AOUT 1944 (IVRY)
                          1107
 INSURRECTION AOUT 1944 (IVRY)
                     0.9193723
```

1.2. Visualization of the data

1.2.1. Loading of 9 random stations throughout the week

In [9]:
```
ngraph = 9
options(repr.plot.width = 50, repr.plot.height = 30)

timeTick = 1 + 24*(0:6)  # vector corresponding to the beginning of days

# select ngraph stations
stations = sample.int(nrow(loading), ngraph-2)
stations = c(997, stations, 1107)

df = melt(loading[stations,])  #the function melt reshapes it from wide to l

p = list()
for (i in 1:ngraph){
  if (velib$bonus[stations[i]]==1) {
      hill= "Hill"
    }else{
      hill="Flat ground"
  }

    dfi = df[df$X1 == velib$names[stations[i]],]
    p[[i]] = ggplot(dfi, aes(x=X2, y=value)) +
        geom_line(col="cornflowerblue", linewidth = 2.5) +
        geom_vline(xintercept=timeTick, col="olivedrab3", linetype="dashed",
        labs(title=velib$names[stations[i]], subtitle = paste("Landscape : "
        geom_hline(yintercept=0.5, col = "darkorange2", linewidth = 2) +
        geom_hline(yintercept=mean(dfi$value), col = "#FAC748", linewidth =
        geom_hline(yintercept=mean(loading), col = "brown1", linewidth = 2)

}
do.call("grid.arrange", c(p, ncol=3))
# pretty pink f88dad
```

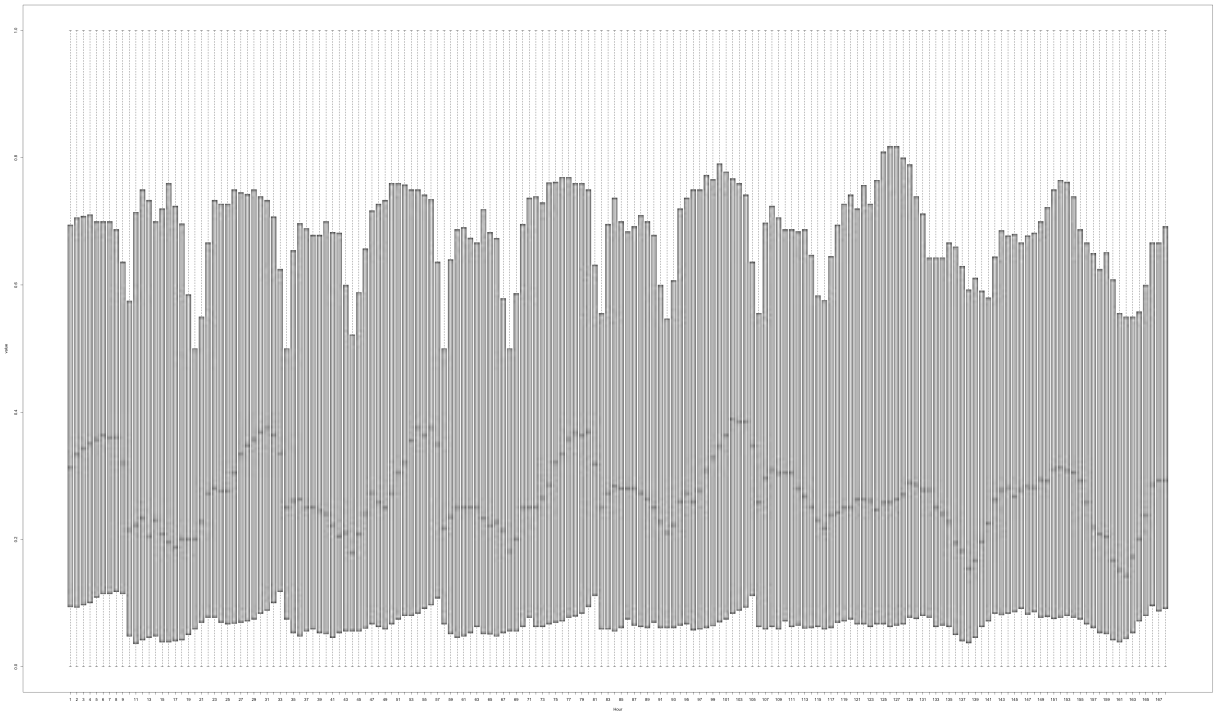
```
Warning message in type.convert.default(X[[i]], ...):
“'as.is' doit être spécifié par l’appelant ; utilisation de TRUE”
Warning message in type.convert.default(X[[i]], ...):
“'as.is' doit être spécifié par l’appelant ; utilisation de TRUE”
```



Blue line : hourly loading per station
Orange line : halfway fullness of a station (ie. loading = 0.5)
Red line : average fullness of all of the stations
Yellow line : average loading of each individual station
Green line : beginning of new day

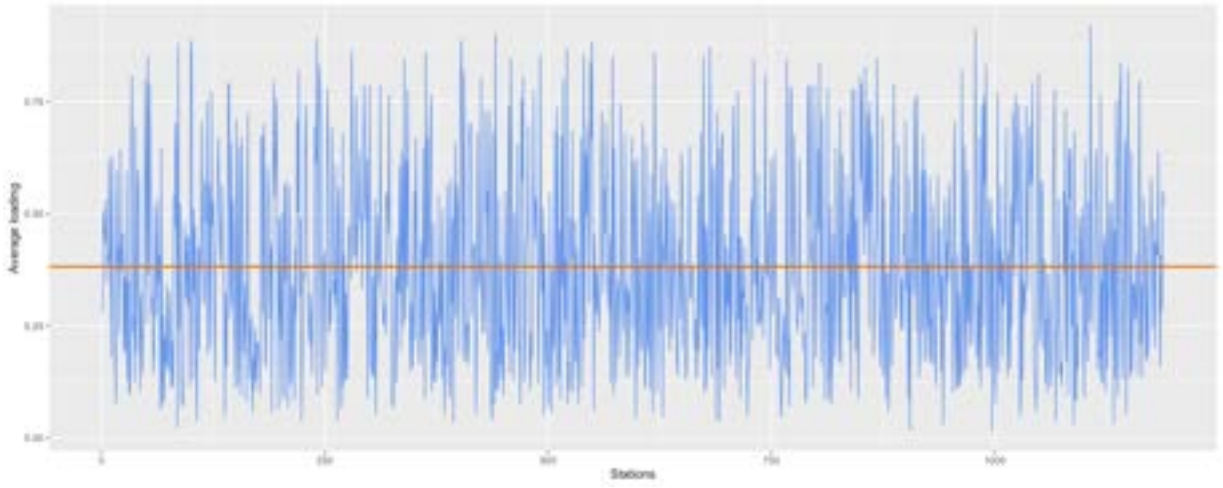
1.2.2. Boxplots of all stations at each hour

In [10]:
```
y = boxplot(loading, xlab= "Hour", ylab = "value") + geom_vline(xintercept=t
```



1.2.3. Average station fill rate

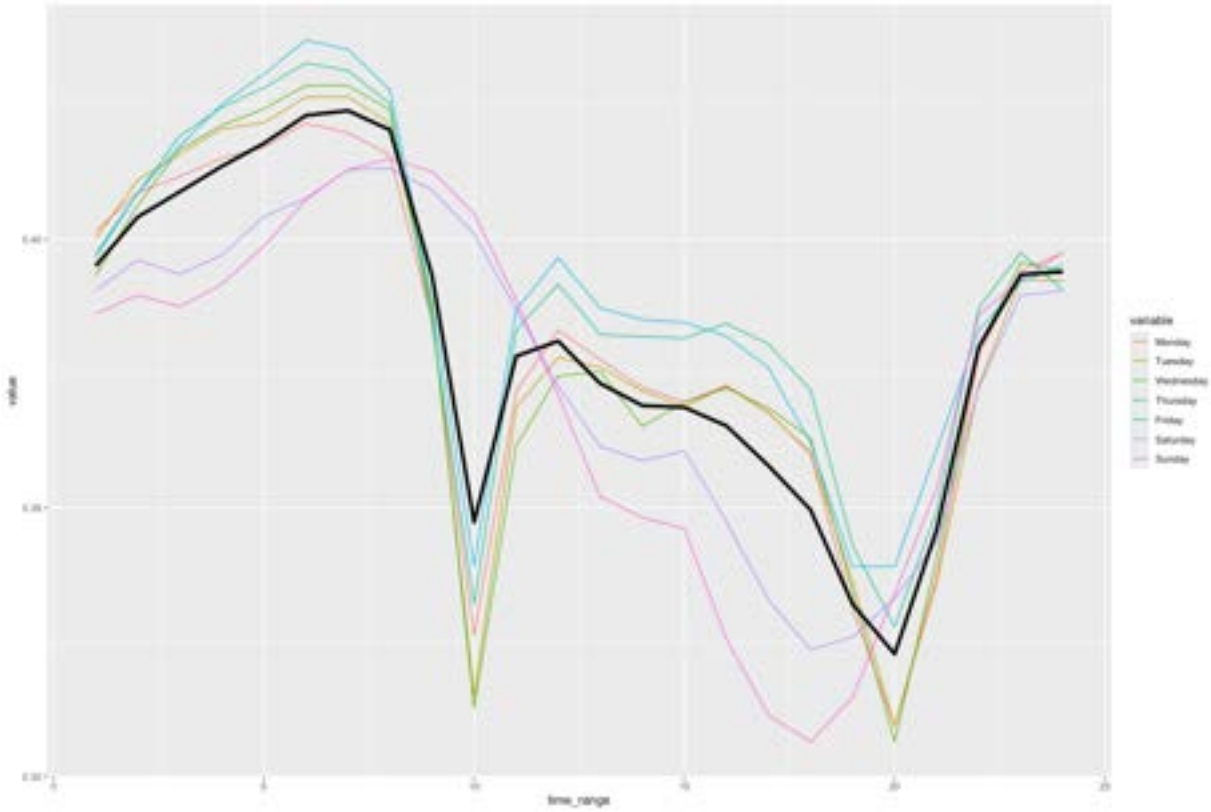
In [11]:
```
options(repr.plot.width = 15, repr.plot.height = 6)

df = data.frame(stations = c(1:nrow(loading)), mean = rowMeans(loading))
ggplot(df, aes(x = stations, y= mean)) +
    geom_line(color = 'cornflowerblue', linewidth=0.5) +
    geom_hline(yintercept = mean(loading), color = 'darkorange2', linewidth=
    labs(x = "Stations", y = "Average loading")
```



Hourly loading for each day

In [12]:
```
mean_per_hour_per_day = colMeans(loading)
mean_per_hour_per_day = matrix(mean_per_hour_per_day, nrow = 24)
mean_per_hour         = rowMeans(mean_per_hour_per_day)
```

```
# --- #

mean_per_hour_per_day            = as.data.frame(mean_per_hour_per_day)
colnames(mean_per_hour_per_day)  = list("Monday", "Tuesday", "Wednesday","Th
mean_per_hour_per_day$time_range = c(1:24)
mean_per_hour_per_day            = melt(mean_per_hour_per_day, id='time_rang

mean_per_hour            = as.data.frame(mean_per_hour)
colnames(mean_per_hour)  = list("Weekly")
mean_per_hour$time_range = c(1:24)

# --- #

options(repr.plot.width = 15, repr.plot.height = 10)

ggplot() +
    geom_line(data=mean_per_hour_per_day, aes(x=time_range, y=value, color=v
    geom_line(data=mean_per_hour, aes(x=time_range, y=Weekly), linewidth = 1
```



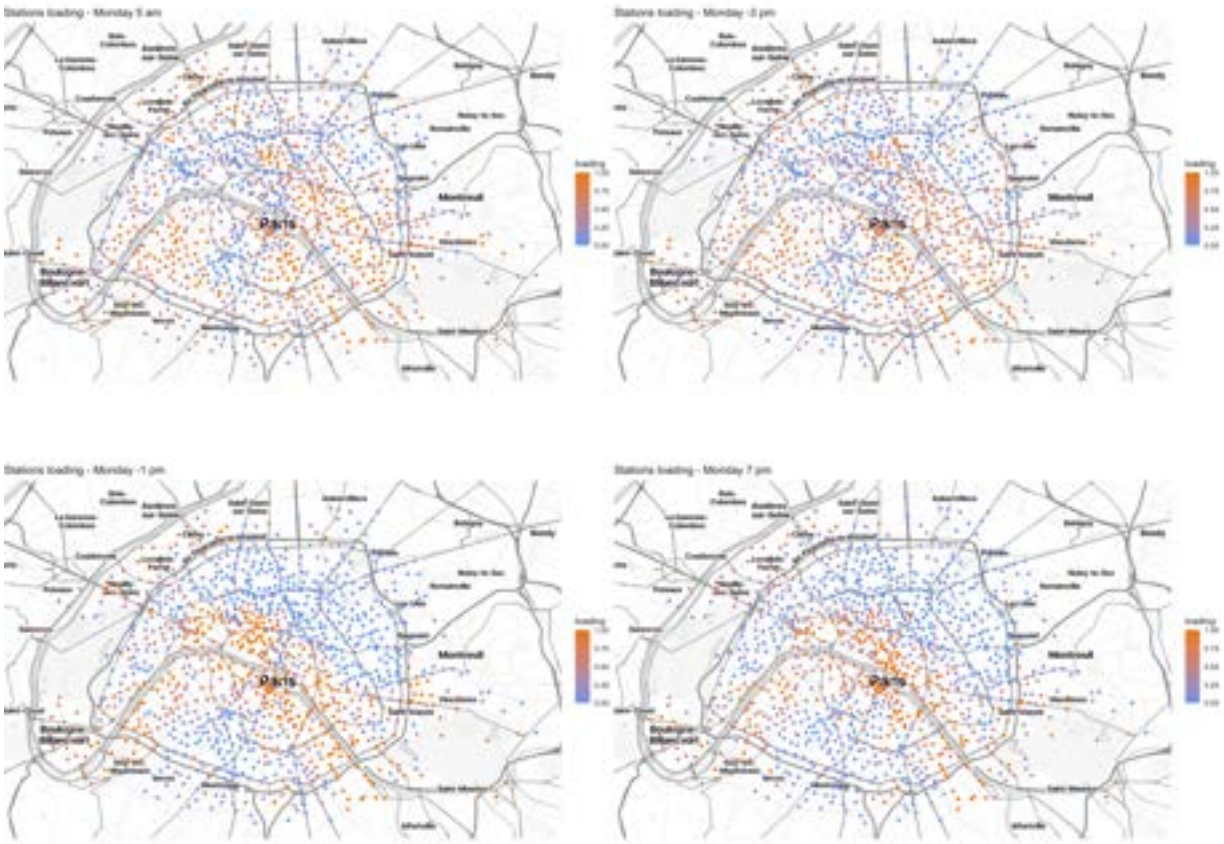
1.3. Visualization of the data on a map of Paris

1.3.1. Stations loading on Monday

In [13]:
```
options(repr.plot.width = 20, repr.plot.height = 15)
hours = c(5, 9, 11, 19)

dfi = coord
p = list()
for (i in 1:length(hours)){
```

```
    dfi$loading = loading[,hours[i]]
    if (hours[i] == 5 || hours[i] == 10){
        p[[i]] = qmplot(data=dfi, longitude, latitude, color=loading) +
        scale_colour_gradientn(colours=c("cornflowerblue", "darkorange2")) +
        labs(title = paste("Stations loading - Monday",hours[i],"am"))
    }
    else{
        p[[i]] = qmplot(data=dfi, longitude, latitude, color=loading) +
        scale_colour_gradientn(colours=c("cornflowerblue", "darkorange2")) +
        labs(title = paste("Stations loading - Monday",hours[i]-12,"pm"))
        }
}

do.call(grid.arrange,c(p, ncol=2))
```

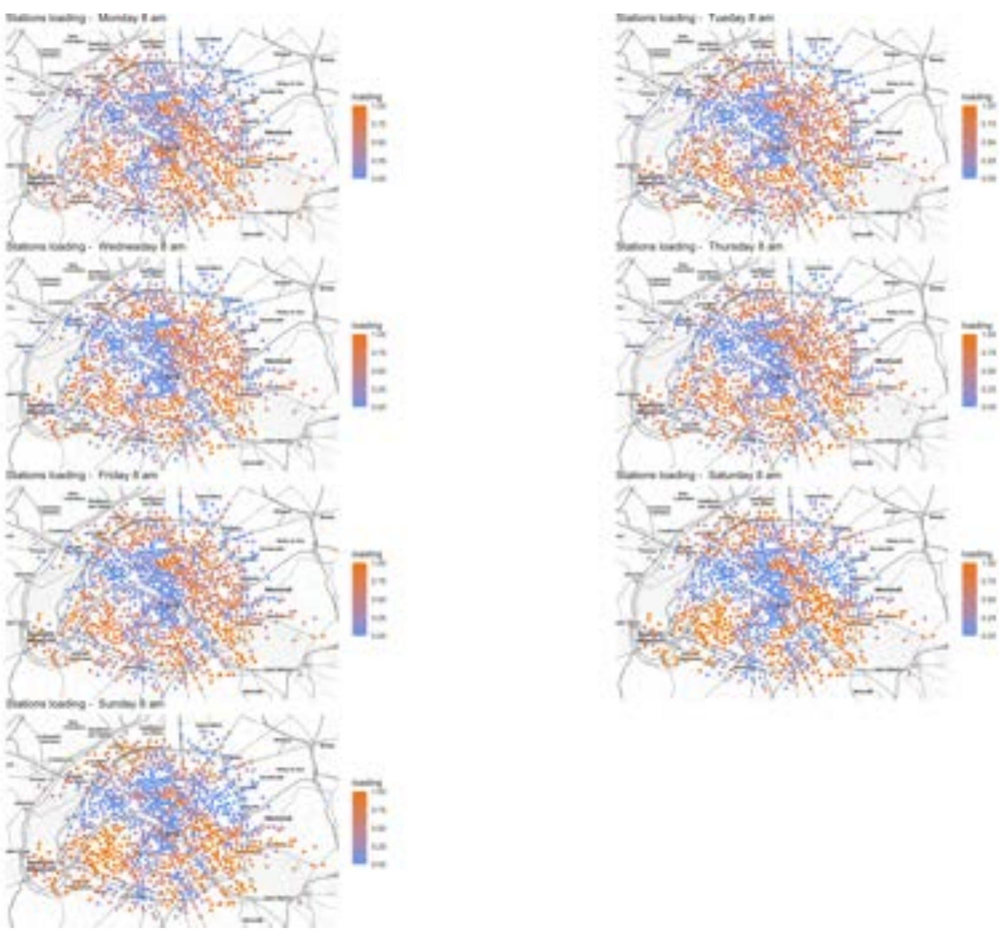
```
ℹ Using `zoom = 12`

ℹ © Stadia Maps © Stamen Design © OpenMapTiles © OpenStreetMap contributors.

ℹ Using `zoom = 12`

ℹ © Stadia Maps © Stamen Design © OpenMapTiles © OpenStreetMap contributors.

ℹ Using `zoom = 12`

ℹ © Stadia Maps © Stamen Design © OpenMapTiles © OpenStreetMap contributors.

ℹ Using `zoom = 12`

ℹ © Stadia Maps © Stamen Design © OpenMapTiles © OpenStreetMap contributors.
```



1.3.2. Stations loading at 8 am

In [14]:
```
options(repr.plot.width = 20, repr.plot.height = 15)
days = c('Monday', 'Tueday', 'Wednesday', 'Thursday', 'Friday', 'Saturday',
dfi = coord
p = list()
for (i in 1:7){
    dfi$loading = loading[,8+(i-1)*24]
    p[[i]] = qmplot(data=dfi, longitude, latitude, color=loading) +
    scale_colour_gradientn(colours=c("cornflowerblue", "darkorange2")) +
    labs(title = paste("Stations loading - ", days[i], "8 am"))
}

do.call(grid.arrange,c(p, ncol=2))
```

```
ℹ Using `zoom = 12`
ℹ © Stadia Maps © Stamen Design © OpenMapTiles © OpenStreetMap contributors.
ℹ Using `zoom = 12`
ℹ © Stadia Maps © Stamen Design © OpenMapTiles © OpenStreetMap contributors.
ℹ Using `zoom = 12`
ℹ © Stadia Maps © Stamen Design © OpenMapTiles © OpenStreetMap contributors.
ℹ Using `zoom = 12`
ℹ © Stadia Maps © Stamen Design © OpenMapTiles © OpenStreetMap contributors.
ℹ Using `zoom = 12`
ℹ © Stadia Maps © Stamen Design © OpenMapTiles © OpenStreetMap contributors.
ℹ Using `zoom = 12`
ℹ © Stadia Maps © Stamen Design © OpenMapTiles © OpenStreetMap contributors.
ℹ Using `zoom = 12`
ℹ © Stadia Maps © Stamen Design © OpenMapTiles © OpenStreetMap contributors.
```

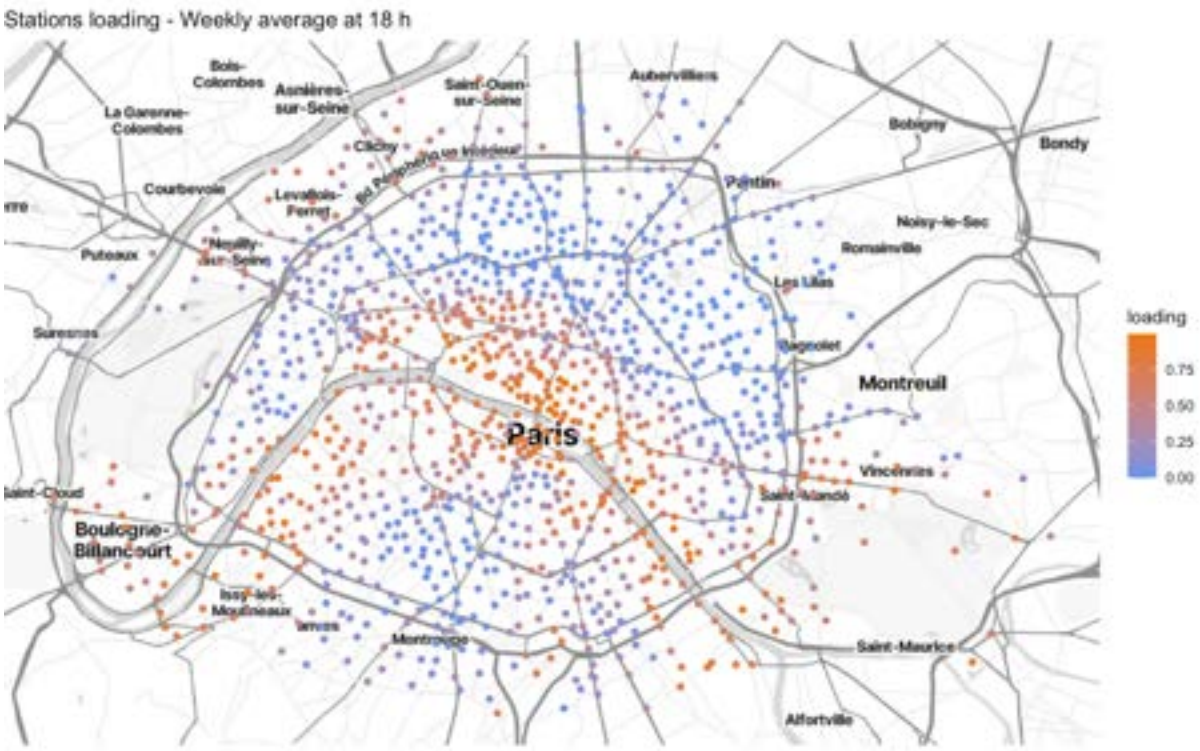


1.3.3. Average loading of all stations at 6 pm

In [15]:
```
h     = 18
hours = seq(h, 168, 24)
load_per_hour = rowMeans(loading[,hours])

df = coord
df$loading = load_per_hour

# --- #

options(repr.plot.width = 10, repr.plot.height = 10)


qmplot(data=df, longitude, latitude, color=loading) +
    scale_colour_gradientn(colours=c("cornflowerblue", "darkorange2")) +
    labs(title = paste('Stations loading - Weekly average at',h,'h'))
```

```
ℹ Using `zoom = 12`
```

```
ℹ © Stadia Maps © Stamen Design © OpenMapTiles © OpenStreetMap contributors.
```



1.4 Influence of Altitude Difference on Station Loading

1.4.1. Pie chart of type of ground and map with stations colored by thir placement

In [16]:
```
#hill = as.factor(velib$bonus)
coord$hill = as.factor(coord$bonus)

options(repr.plot.width = 20, repr.plot.height = 10)

df = data.frame(size=c(sum(coord$bonus==0), sum(coord$bonus==1)),
                labels = c('No hill','Hill'))

plot1 = qmplot(data=coord, longitude, latitude, color=hill) +
    scale_color_manual(values = c("0" = "cornflowerblue", "1" = "darkorange2
    labs(title = 'Hilltop stations')
```

```
plot2 = ggplot(df, aes(x="", y=size, fill=labels)) +
    geom_bar(stat="identity", width=1) +
    geom_text(aes(label=size), position = position_stack(vjust = 0.5)) +
    coord_polar(theta = "y") +
    scale_fill_manual(values = c("No hill" = "cornflowerblue", "Hill" = "dar
    theme_void()

grid.arrange(plot1, plot2, ncol=2)
```

```
ℹ Using `zoom = 12`
```

```
ℹ © Stadia Maps © Stamen Design © OpenMapTiles © OpenStreetMap contributors.
```

We can see that only around 10% of all the stations in the data set are located on hills.

1.4.2. Average fullnes of stations on hills and flat ground

In [17]:
```
hilltop_average = 0
flat_average = 0
compteur_hill = 0
compteur_flat = 0

for (i in 1:nrow(loading_hill)){
    if (loading_hill[i, 169] == 1){
        hilltop_average = hilltop_average + mean(loading[i])
        compteur_hill = compteur_hill + 1
    }
    else{
        flat_average = flat_average + mean(loading[i])
        compteur_flat = compteur_flat + 1
    }
}

hilltop_average = hilltop_average / compteur_hill
flat_average = flat_average / compteur_flat
```

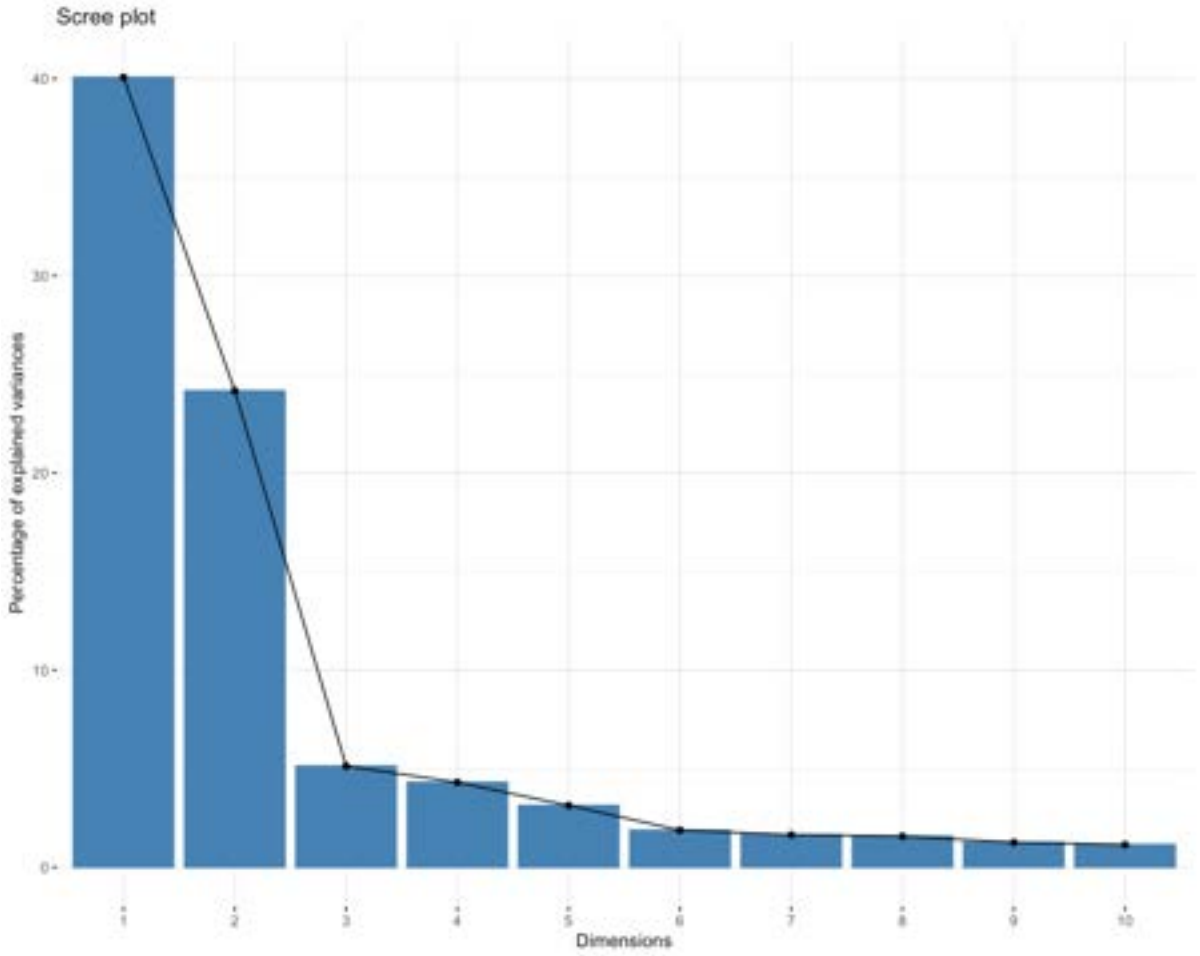
```
cat('The average fullness on hilltop stations is', hilltop_average, 'whereas
```

```
The average fullness on hilltop stations is 0.1218327 whereas the average fu
llness of flatground stations is  0.4349479
```

Part 2 : Principal Component Analysis (PCA)

2.1. How many components on the PCA should we keep?

In [18]:
```
options(repr.plot.width = 10, repr.plot.height = 8)

v2 = data.frame(loading, hill = as.factor(velib$bonus))
#pca = PCA(loading, quali.sup = velib$bonus, scale.unit = TRUE, graph=FALSE)

pca = PCA(v2, ncp = 6, scale.unit = FALSE, quali.sup= 169, graph = FALSE)

fviz_eig(pca)
```



In [19]:
```
head(pca$eig, 10)
```

A matrix: 10 × 3 of type dbl

	eigenvalue	percentage of variance	cumulative percentage of variance
comp 1	7.9730841	40.071667	40.07167
comp 2	4.8089597	24.169195	64.24086
comp 3	1.0215417	5.134133	69.37500
comp 4	0.8574387	4.309373	73.68437
comp 5	0.6236212	3.134238	76.81861
comp 6	0.3739460	1.879403	78.69801
comp 7	0.3280678	1.648825	80.34684
comp 8	0.3134050	1.575132	81.92197
comp 9	0.2507254	1.260113	83.18208
comp 10	0.2283967	1.147891	84.32997

We decided to keep six main components to explain 78% of the variance.

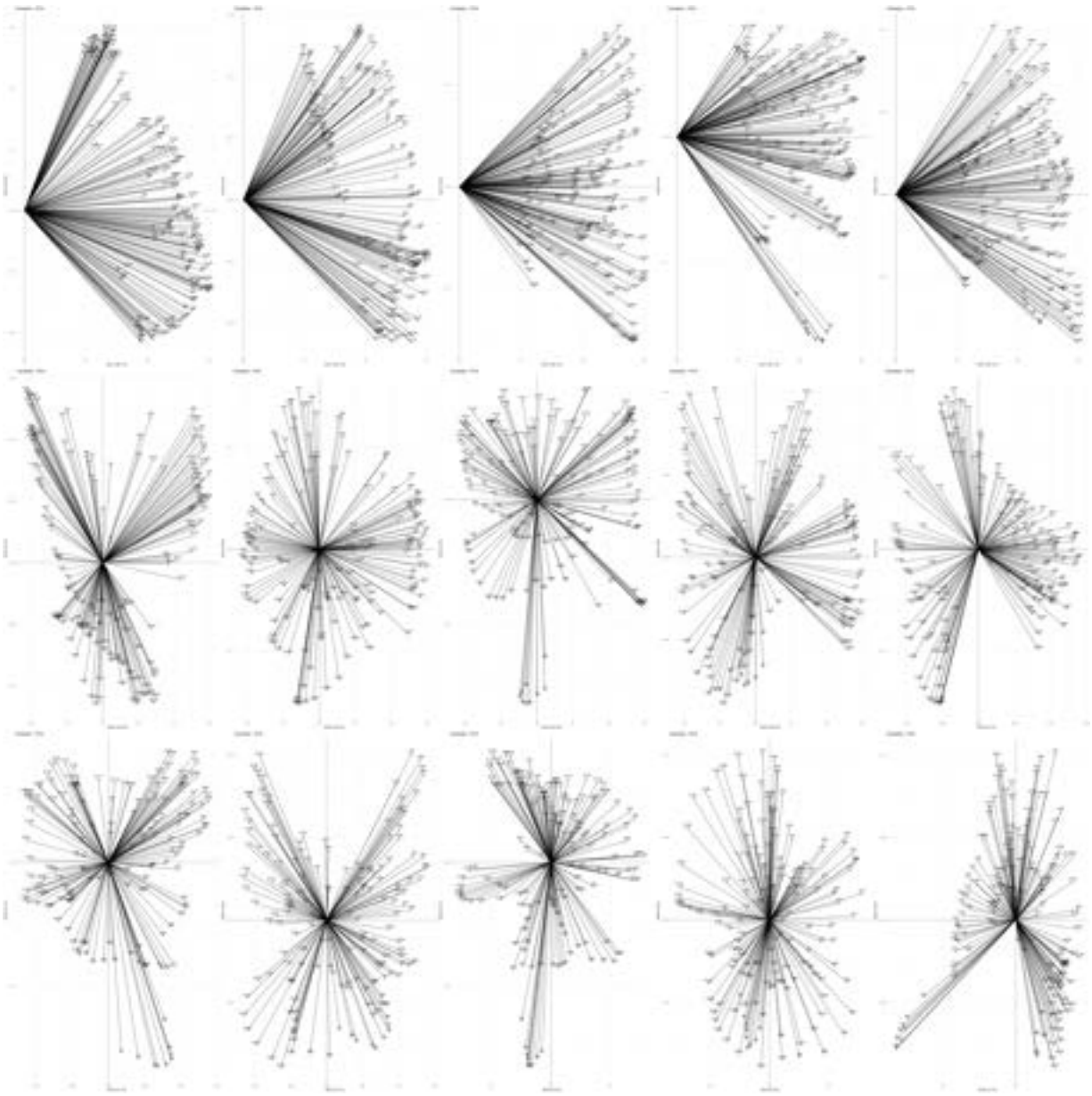
2.2. Visualization the dispersion of the observations on the 6-th first components of the PCA

2.3. Variable factor maps

2.3.1. Variable Factor Maps for the first 6 principal components

In [20]:
```
options(repr.plot.width=50, repr.plot.height = 50)

p1 = fviz_pca_var(pca,axes=c(1,2))
p2 = fviz_pca_var(pca,axes=c(1,3))
p3 = fviz_pca_var(pca,axes=c(1,4))
p4 = fviz_pca_var(pca,axes=c(1,5))
p5 = fviz_pca_var(pca,axes=c(1,6))

p6 = fviz_pca_var(pca,axes=c(2,3))
p7 = fviz_pca_var(pca,axes=c(2,4))
p8 = fviz_pca_var(pca,axes=c(2, 5))
p9 = fviz_pca_var(pca,axes=c(2,6))

p10 = fviz_pca_var(pca,axes=c(3,4))
p11 = fviz_pca_var(pca,axes=c(3,5))
p12 = fviz_pca_var(pca,axes=c(3,6))

p13 = fviz_pca_var(pca,axes=c(4,5))
p14 = fviz_pca_var(pca,axes=c(4,6))

p15 = fviz_pca_var(pca,axes=c(5,6))
```

```
gridExtra::grid.arrange(p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, p11, p12, p
```



2.3.2. Variable Factor Map labeled by hill position

After further reflection we realised that coloring a variable (ie. an hour of the week) depending on wheather or not it is on a hill is not pertinent so we will not do it in the R notebook.

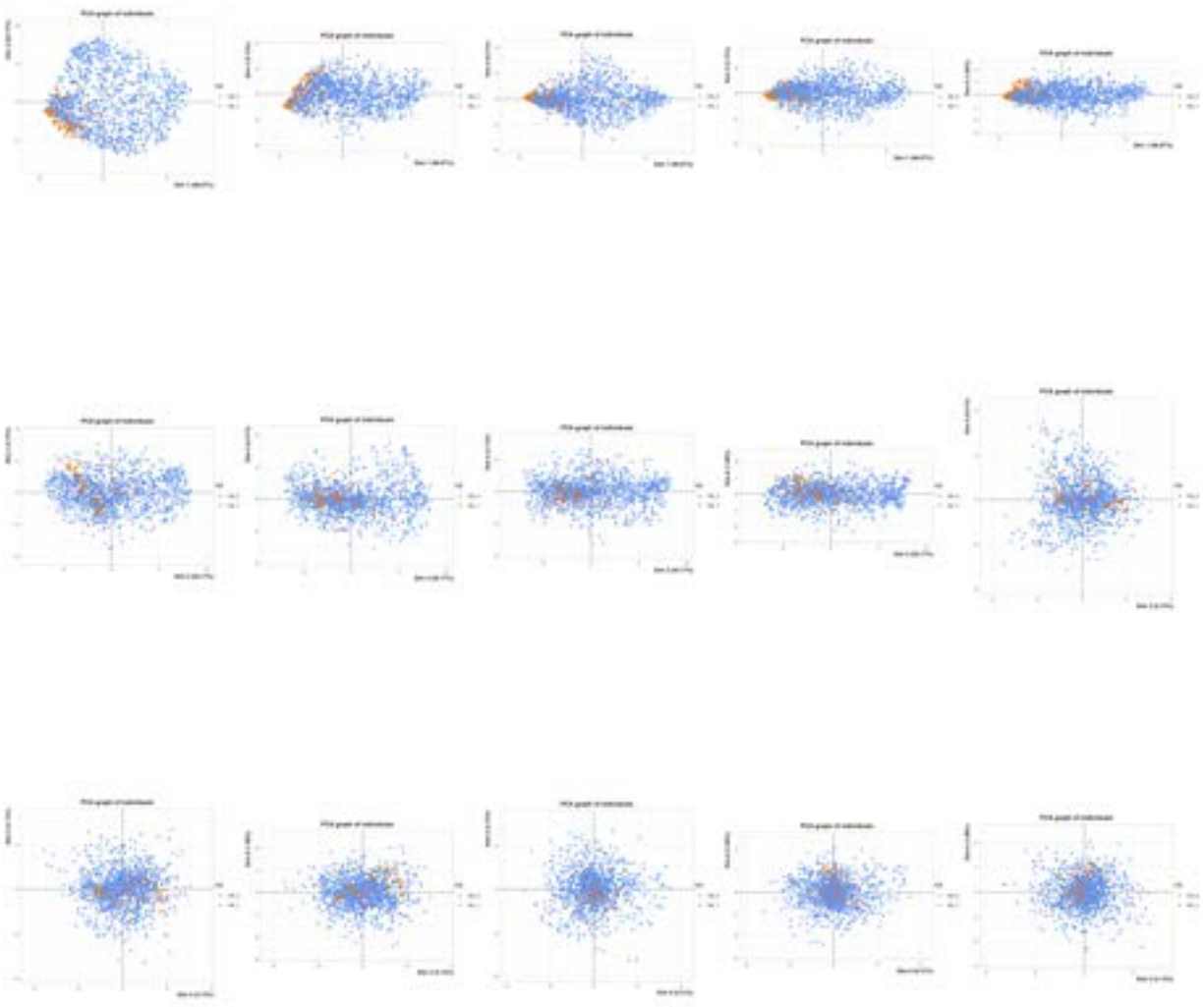
2.3.3. Variable Factor Map labeled by day and night hours

We were unable to label the variables by day and night in R. The results are available in the python notebook

2.4. Individual factor maps

2.4.1. Individual factor maps colored by hill position

In [21]:
```
options(repr.plot.width=30, repr.plot.height = 30)

p1 = plot(pca, axes = c(1,2), choix = "ind", habillage = 169, label = "none"
p2 = plot(pca, axes = c(1,3), choix = "ind", habillage = 169, label = "none"
p3 = plot(pca, axes = c(1,4), choix = "ind", habillage = 169, label = "none"
p4 = plot(pca, axes = c(1,5), choix = "ind", habillage = 169, label = "none"
p5 = plot(pca, axes = c(1,6), choix = "ind", habillage = 169, label = "none"

p6 = plot(pca, axes = c(2,3), choix = "ind", habillage = 169, label = "none"
p7 = plot(pca, axes = c(2,4), choix = "ind", habillage = 169, label = "none"
p8 = plot(pca, axes = c(2,5), choix = "ind", habillage = 169, label = "none"
p9 = plot(pca, axes = c(2,6), choix = "ind", habillage = 169, label = "none"

p10 = plot(pca, axes = c(3,4), choix = "ind", habillage = 169, label = "none
p11 = plot(pca, axes = c(3,5), choix = "ind", habillage = 169, label = "none
p12 = plot(pca, axes = c(3,6), choix = "ind", habillage = 169, label = "none

p13 = plot(pca, axes = c(4,5), choix = "ind", habillage = 169, label = "none
p14 = plot(pca, axes = c(4,6), choix = "ind", habillage = 169, label = "none

p15 = plot(pca, axes = c(5,6), choix = "ind", habillage = 169, label = "none

gridExtra::grid.arrange(p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, p11, p12, p
```



2.4.2. Individual factor maps colored by day and night

After further reflection we realised that coloring an individual (ie. a velib station) depending on wheather it is night or day is not pertinent so we will not do it in the R notebook.

2.4.3. Individual factor maps of mean of all stations

We were unable to label the variables by day and night in R. The results are available in the python notebook

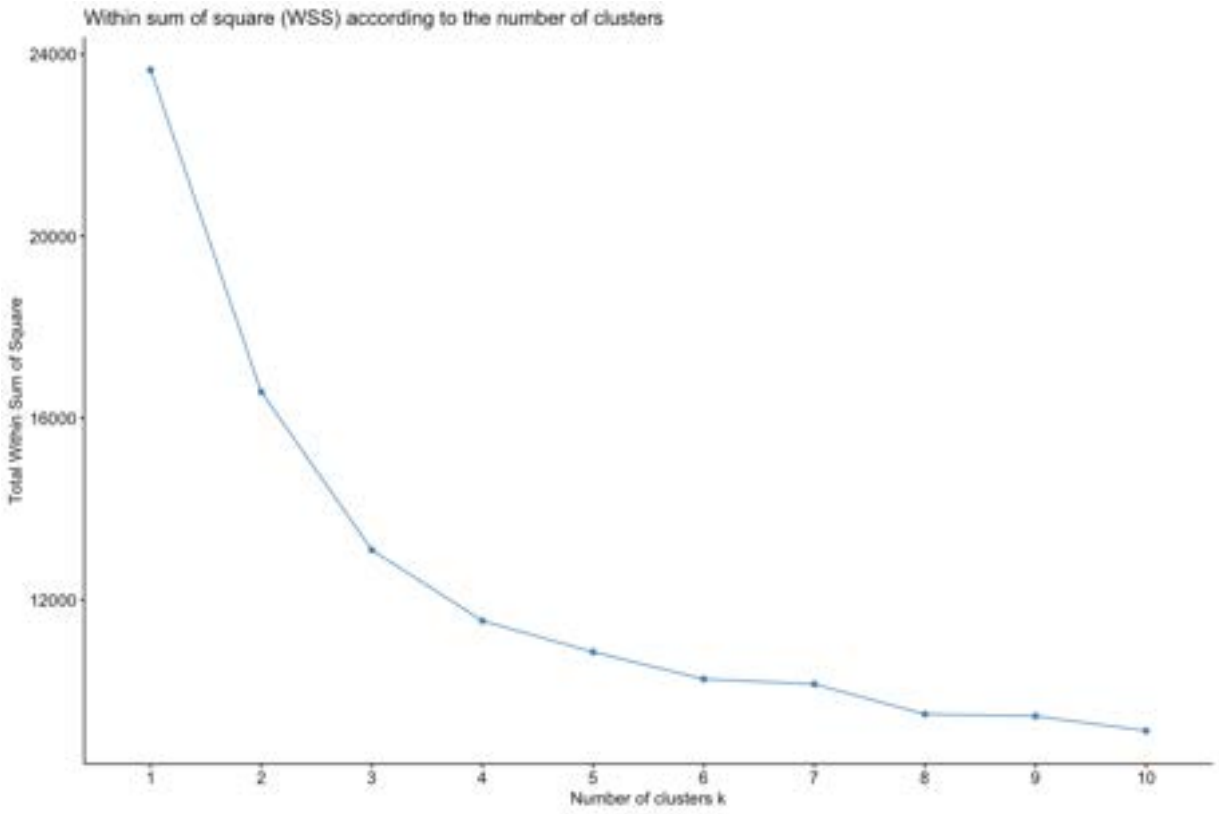
Part 3: clustering on original data

3.1. K-means clustering

3.1.1. Selection of the number of clusters

3.1.1.1. Determining the number of clusters using the total within sum of square metric

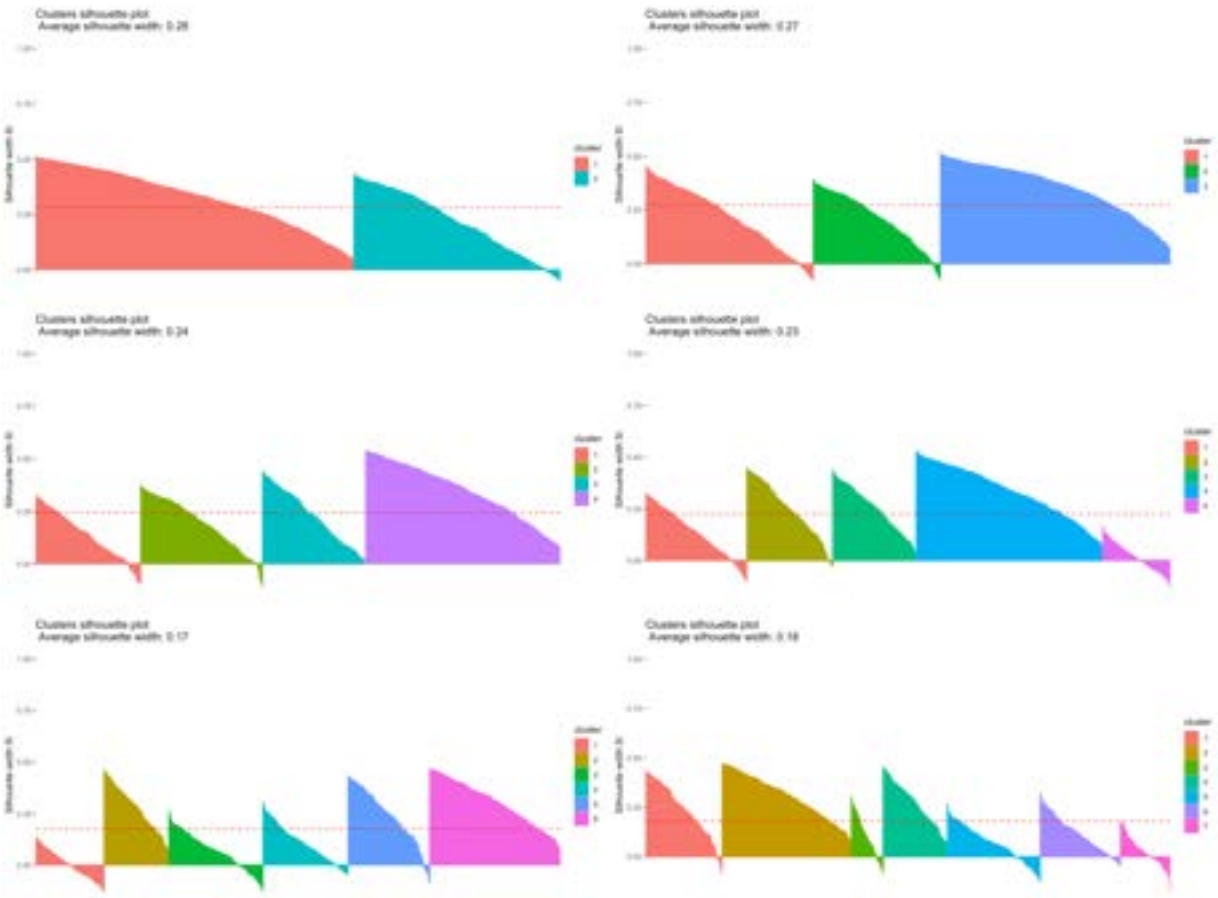
In [22]:
```
options(repr.plot.width = 12, repr.plot.height = 8)

fviz_nbclust(loading, FUNcluster=kmeans, method="wss") +
    ggtitle("Within sum of square (WSS) according to the number of clusters"
```



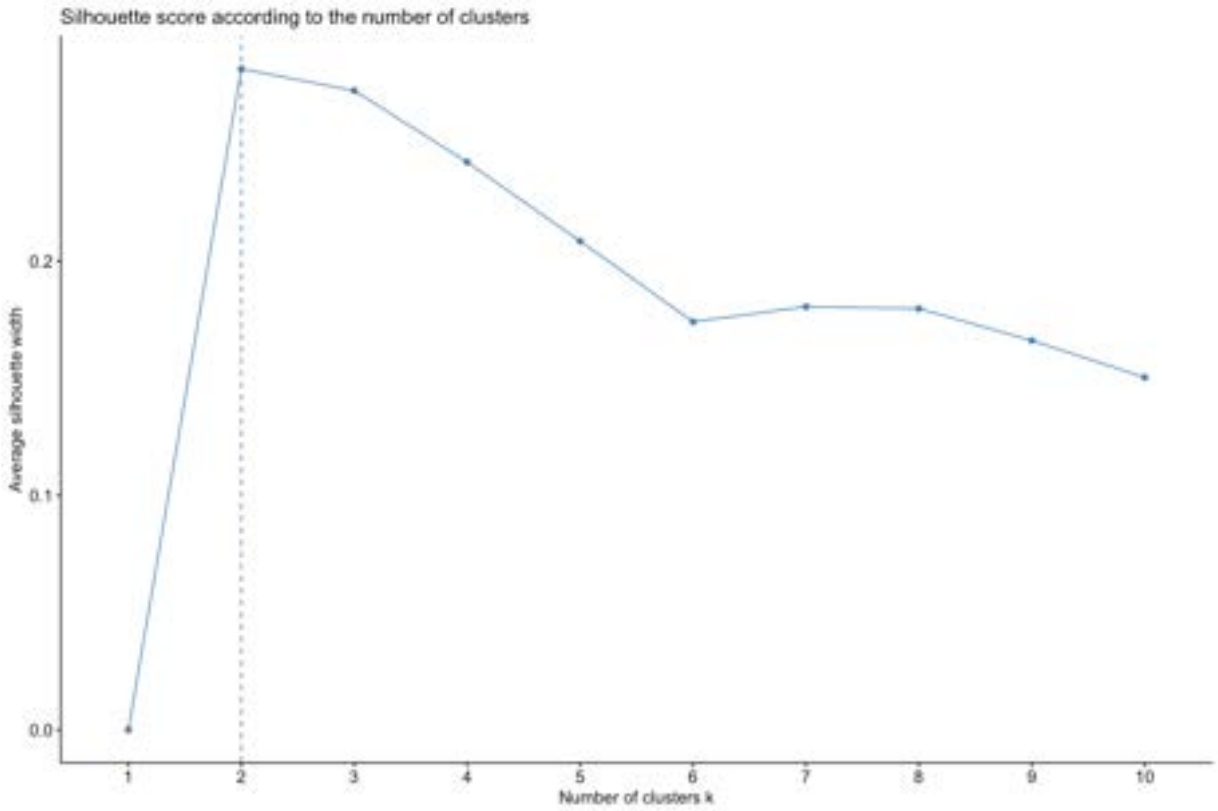
3.1.1.2. Determining the number of clusters using the silhouette scores metric

In [23]:
```
# Silhouette plots, according to the number of clusters
options(repr.plot.width = 20, repr.plot.height = 15)

reskmeans2 = kmeans(loading, centers=2)
reskmeans3 = kmeans(loading, centers=3)
reskmeans4 = kmeans(loading, centers=4)
reskmeans5 = kmeans(loading, centers=5)
reskmeans6 = kmeans(loading, centers=6)
reskmeans7 = kmeans(loading, centers=7)


sil = silhouette(reskmeans2$cluster, dist(loading))
p1 = fviz_silhouette(sil, print.summary = FALSE)
```

```
sil = silhouette(reskmeans3$cluster, dist(loading))
p2 = fviz_silhouette(sil, print.summary = FALSE)

sil = silhouette(reskmeans4$cluster, dist(loading))
p3 = fviz_silhouette(sil, print.summary = FALSE)

sil = silhouette(reskmeans5$cluster, dist(loading))
p4 = fviz_silhouette(sil, print.summary = FALSE)

sil = silhouette(reskmeans6$cluster, dist(loading))
p5 = fviz_silhouette(sil, print.summary = FALSE)

sil = silhouette(reskmeans7$cluster, dist(loading))
p6 = fviz_silhouette(sil, print.summary = FALSE)

grid.arrange(p1,p2,p3,p4,p5,p6, ncol=2)
```



In [24]:
```
options(repr.plot.width = 12, repr.plot.height = 8)

fviz_nbclust(loading, FUNcluster=kmeans, method="silhouette") +
    ggtitle("Silhouette score according to the number of clusters")
```



We will perform the study with k in (2, 3, 4)

3.1.2. Visualization and interpretation of k-means clusters

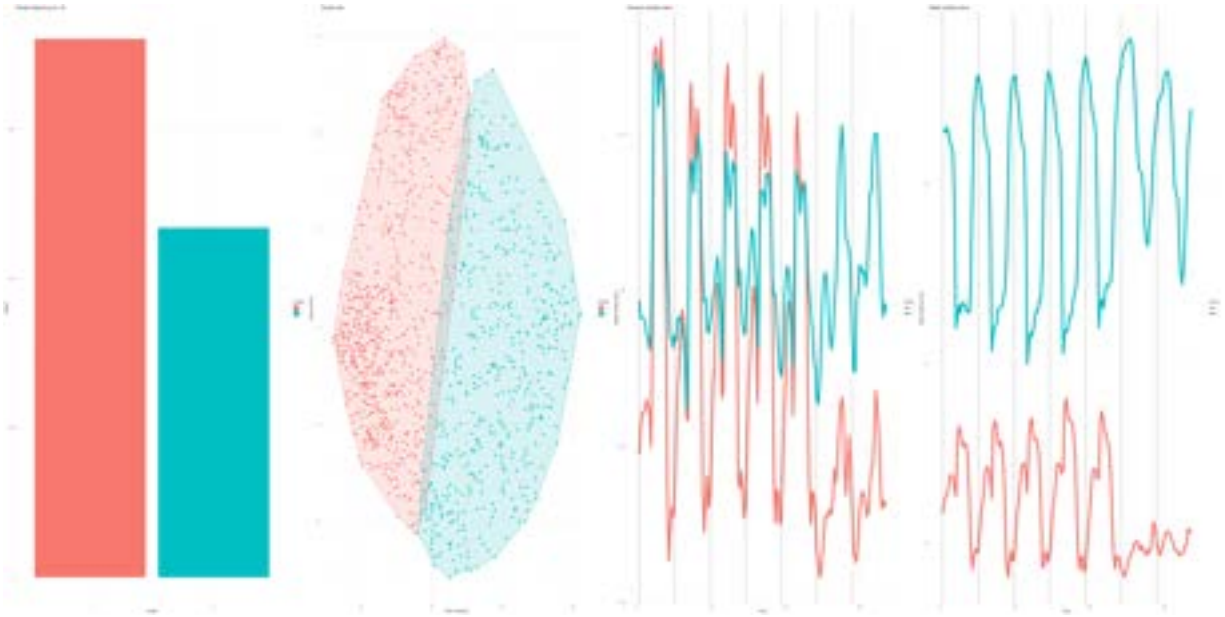
3.1.2.1. Four descriptive plots of cluster

For each value of k we have four different plots:

1. histogram of the frequency of each cluster
2. individual factor map colored by cluster
3. variance of loading scores for each cluster
4. mean loading values per hour cluster

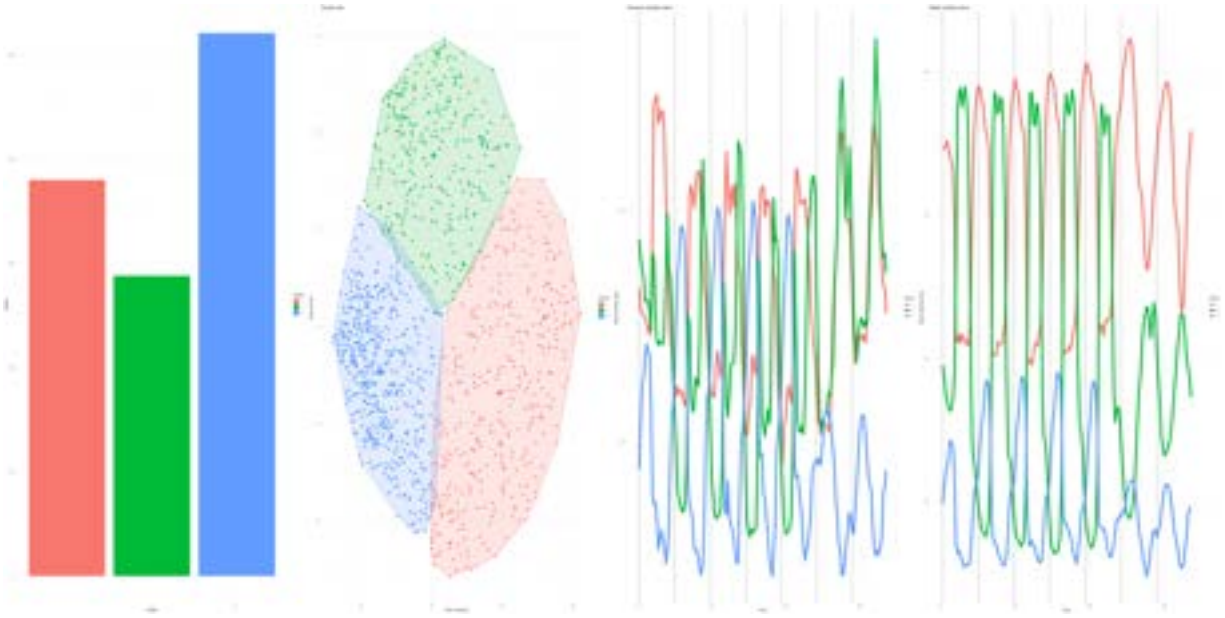
In [25]:
```
time_tick = 1 + 24*(0:6)
options(repr.plot.width = 50, repr.plot.height = 25)

###############
#### k = 2 ####
###############

df = data.frame(cluster = c("1", "2"), effectif = c(reskmeans2$size))

load1 = loading[reskmeans2$cluster==1,]
load2 = loading[reskmeans2$cluster==2,]

meanload1 = colMeans(load1)
meanload2 = colMeans(load2)
dfmean = as.data.frame(meanload1)
```

```
dfmean$meanload2 = meanload2
time_range = 1:ncol(loading)
dfmean$time_range = time_range

varload1 = colVars(load1)
varload2 = colVars(load2)
dfvar = as.data.frame(varload1)
dfvar$varload2 = varload2
time_range = 1:ncol(loading)
dfvar$time_range = time_range

p1 = ggplot(df, aes(x = cluster, y = effectif, fill=cluster)) +
    geom_bar(stat='identity') +
    labs(title = "Cluster frequency (k = 2)") +
    theme_minimal()

p2 = fviz_cluster(reskmeans2, data=velib$data, ellipse.type="convex", labels

p3 = ggplot(dfvar,aes(x=time_range)) +
    geom_line(aes(y=varload1, color='1'), linewidth = 3) +
    geom_line(aes(y=varload2, color='2'), linewidth = 3) +
    labs(title = "Variance loading value", x = "Hour", y = "Mean loading val
    geom_vline(xintercept=time_tick, linetype="dashed") +
    theme_minimal()

p4 = ggplot(dfmean,aes(x=time_range)) +
    geom_line(aes(y=meanload1, color='1'), linewidth = 3) +
    geom_line(aes(y=meanload2, color='2'), linewidth = 3) +
    labs(title = "Mean loading value", x = "Hour", y = "Mean loading value")
    geom_vline(xintercept=time_tick, linetype="dashed") +
    theme_minimal()

ggarrange(p1,p2,p3,p4, ncol = 4)
```



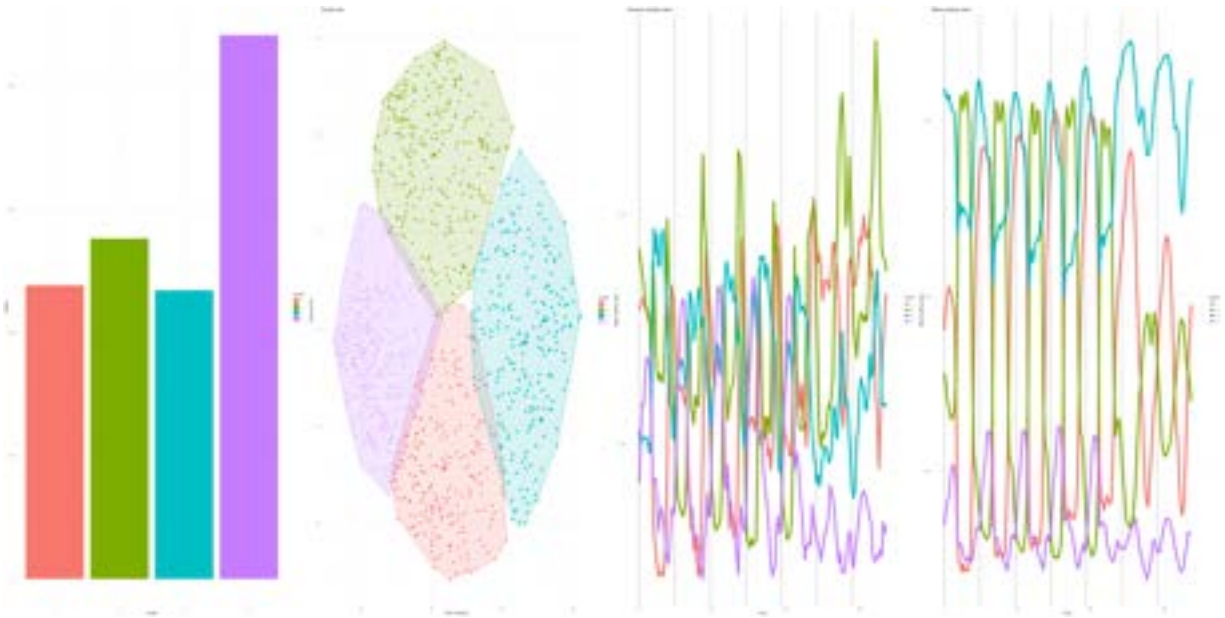
In [26]:
```
###############
#### k = 3 ####
###############
```

```
df = data.frame(cluster = c("1", "2", "3"), effectif = c(reskmeans3$size))

load1 = loading[reskmeans3$cluster==1,]
load2 = loading[reskmeans3$cluster==2,]
load3 = loading[reskmeans3$cluster==3,]

meanload1 = colMeans(load1)
meanload2 = colMeans(load2)
meanload3 = colMeans(load3)
dfmean = as.data.frame(meanload1)
dfmean$meanload2 = meanload2
dfmean$meanload3 = meanload3
time_range = 1:ncol(loading)
dfmean$time_range = time_range

varload1 = colVars(load1)
varload2 = colVars(load2)
varload3 = colVars(load3)
dfvar = as.data.frame(varload1)
dfvar$varload2 = varload2
dfvar$varload3 = varload3
time_range = 1:ncol(loading)
dfvar$time_range = time_range

p5 = ggplot(df, aes(x = cluster, y = effectif, fill=cluster)) +
    geom_bar(stat='identity') +
    theme_minimal()

p6 = fviz_cluster(reskmeans3, data=velib$data, ellipse.type="convex", labels

p7 = ggplot(dfvar,aes(x=time_range)) +
    geom_line(aes(y=varload1, color='1'), linewidth = 3) +
    geom_line(aes(y=varload2, color='2'), linewidth = 3) +
    geom_line(aes(y=varload3, color='3'), linewidth = 3) +
    labs(title = "Variance loading value", x = "Hour", y = "Mean loading val
    geom_vline(xintercept=time_tick, linetype="dashed") +
    theme_minimal()

p8 = ggplot(dfmean,aes(x=time_range)) +
    geom_line(aes(y=meanload1, color='1'), linewidth = 3) +
    geom_line(aes(y=meanload2, color='2'), linewidth = 3) +
    geom_line(aes(y=meanload3, color='3'), linewidth = 3) +
    labs(title = "Mean loading value", x = "Hour", y = "Mean loading value")
    geom_vline(xintercept=time_tick, linetype="dashed") +
    theme_minimal()

ggarrange(p5,p6,p7,p8, ncol = 4)
```



In [27]:
```
###############
#### k = 4 ####
###############

df = data.frame(cluster = c("1", "2", "3", "4"), effectif = c(reskmeans4$siz

load1 = loading[reskmeans4$cluster==1,]
load2 = loading[reskmeans4$cluster==2,]
load3 = loading[reskmeans4$cluster==3,]
load4 = loading[reskmeans4$cluster==4,]

meanload1 = colMeans(load1)
meanload2 = colMeans(load2)
meanload3 = colMeans(load3)
meanload4 = colMeans(load4)
dfmean = as.data.frame(meanload1)
dfmean$meanload2 = meanload2
dfmean$meanload3 = meanload3
dfmean$meanload4 = meanload4
time_range = 1:ncol(loading)
dfmean$time_range = time_range

varload1 = colVars(load1)
varload2 = colVars(load2)
varload3 = colVars(load3)
varload4 = colVars(load4)
dfvar = as.data.frame(varload1)
dfvar$varload2 = varload2
dfvar$varload3 = varload3
dfvar$varload4 = varload4
time_range = 1:ncol(loading)
dfvar$time_range = time_range

p9 = ggplot(df, aes(x = cluster, y = effectif, fill=cluster)) +
    geom_bar(stat='identity') +
    theme_minimal()
```

```
p10 = fviz_cluster(reskmeans4, data=velib$data, ellipse.type="convex", label

p11 = ggplot(dfvar,aes(x=time_range)) +
    geom_line(aes(y=varload1, color='1'), linewidth = 3) +
    geom_line(aes(y=varload2, color='2'), linewidth = 3) +
    geom_line(aes(y=varload3, color='3'), linewidth = 3) +
    geom_line(aes(y=varload4, color='4'), linewidth = 3) +
    labs(title = "Variance loading value", x = "Hour", y = "Mean loading val
    geom_vline(xintercept=time_tick, linetype="dashed") +
    theme_minimal()

p12 = ggplot(dfmean,aes(x=time_range)) +
    geom_line(aes(y=meanload1, color='1'), linewidth = 3) +
    geom_line(aes(y=meanload2, color='2'), linewidth = 3) +
    geom_line(aes(y=meanload3, color='3'), linewidth = 3) +
    geom_line(aes(y=meanload4, color='4'), linewidth = 3) +
    labs(title = "Mean loading value", x = "Hour", y = "Mean loading value")
    geom_vline(xintercept=time_tick, linetype="dashed") +
    theme_minimal()

ggarrange(p9,p10,p11,p12, ncol = 4)
```



3.1.2.2. Visualization of clusters on the map

In [28]:
```
#hill = as.factor(velib$bonus)
options(repr.plot.width = 20, repr.plot.height = 10)

##############

reskmeans2$cluster = as.factor(reskmeans2$cluster)

df2 = data.frame(size=c(sum(reskmeans2$cluster==1), sum(reskmeans2$cluster==
                labels = c('cluster 1','cluster 2'))

p2 = qmplot(data=coord, longitude, latitude, color=reskmeans2$cluster) +
    #scale_color_manual(values = c("1" = "cornflowerblue", "2" = "darkorange
    labs(title = 'Two clusters with kmeans on complete data')
```

```
##############

reskmeans3$cluster = as.factor(reskmeans3$cluster)

df3 = data.frame(size=c(sum(reskmeans3$cluster==1), sum(reskmeans3$cluster==
                labels = c('cluster 1','cluster 2', 'cluster 3'))

p3 = qmplot(data=coord, longitude, latitude, color=reskmeans3$cluster) +
    #scale_color_manual(values = c("1" = "cornflowerblue", "2" = "darkorange
    labs(title = 'Three clusters with kmeans on complete data')

##############

reskmeans4$cluster = as.factor(reskmeans4$cluster)

df4 = data.frame(size=c(sum(reskmeans4$cluster==1), sum(reskmeans4$cluster==
                labels = c('cluster 1','cluster 2', 'cluster 3', 'cluster 4'

p4 = qmplot(data=coord, longitude, latitude, color=reskmeans4$cluster) +
    #scale_color_manual(values = c("1" = "cornflowerblue", "2" = "darkorange
    labs(title = 'Four clusters with kmeans on complete data')

ggpubr::ggarrange(p2,p3, p4)
```

```
ℹ Using `zoom = 12`
```

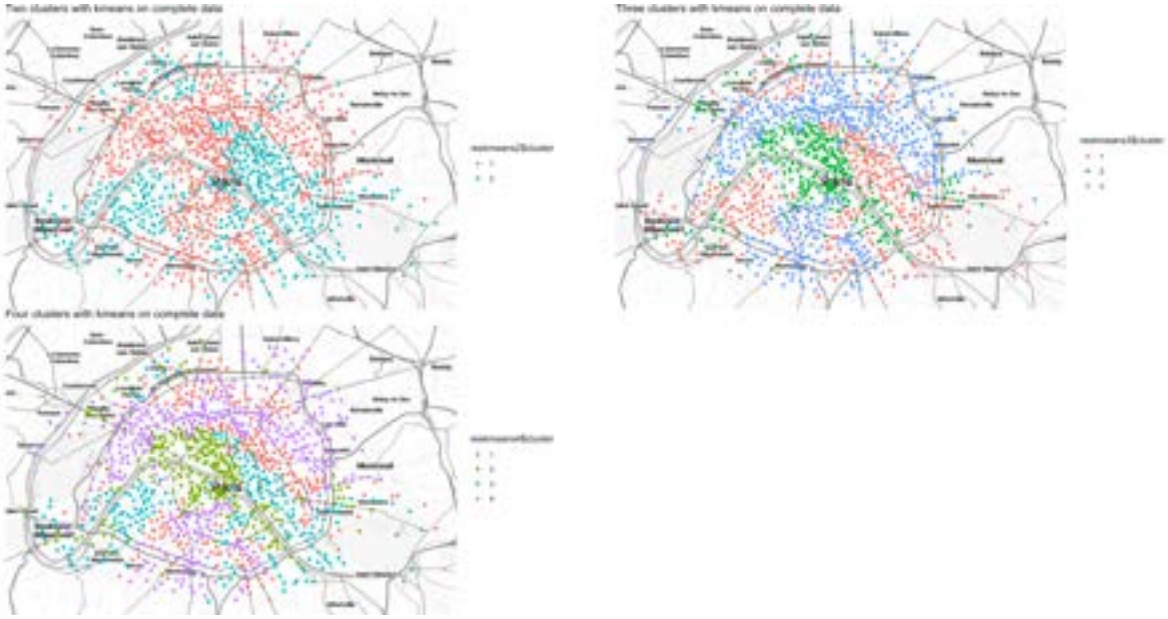
```
ℹ © Stadia Maps © Stamen Design © OpenMapTiles © OpenStreetMap contributors.
```

```
ℹ Using `zoom = 12`
```

```
ℹ © Stadia Maps © Stamen Design © OpenMapTiles © OpenStreetMap contributors.
```

```
ℹ Using `zoom = 12`
```

```
ℹ © Stadia Maps © Stamen Design © OpenMapTiles © OpenStreetMap contributors.
```

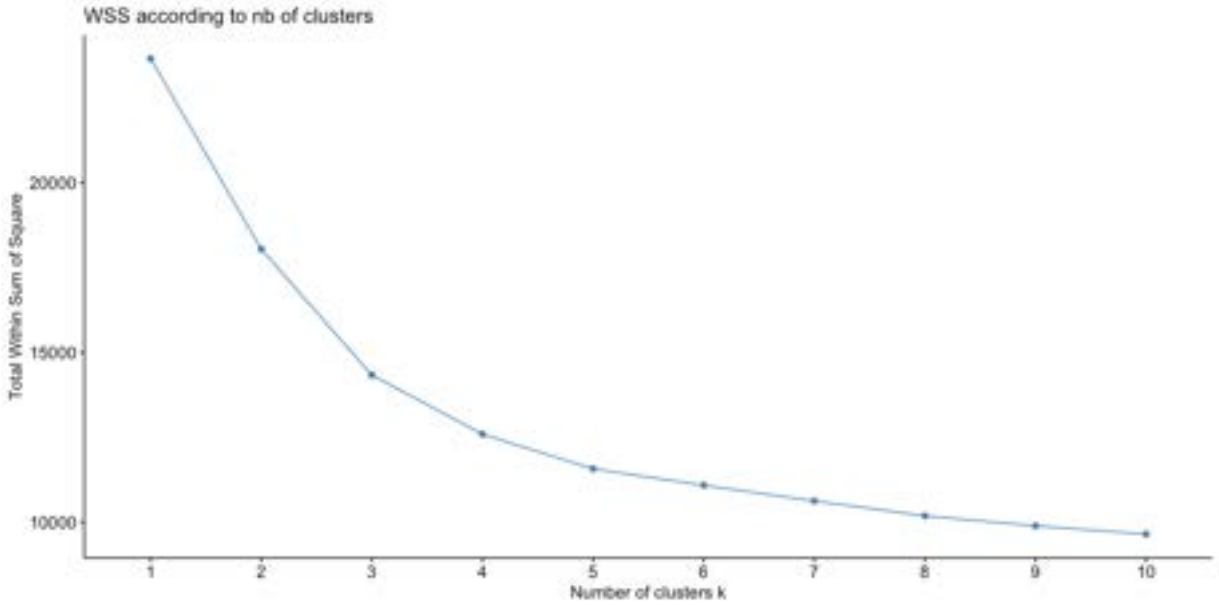


3.2. HCA clustering

3.2.1. Selection of the number of clusters

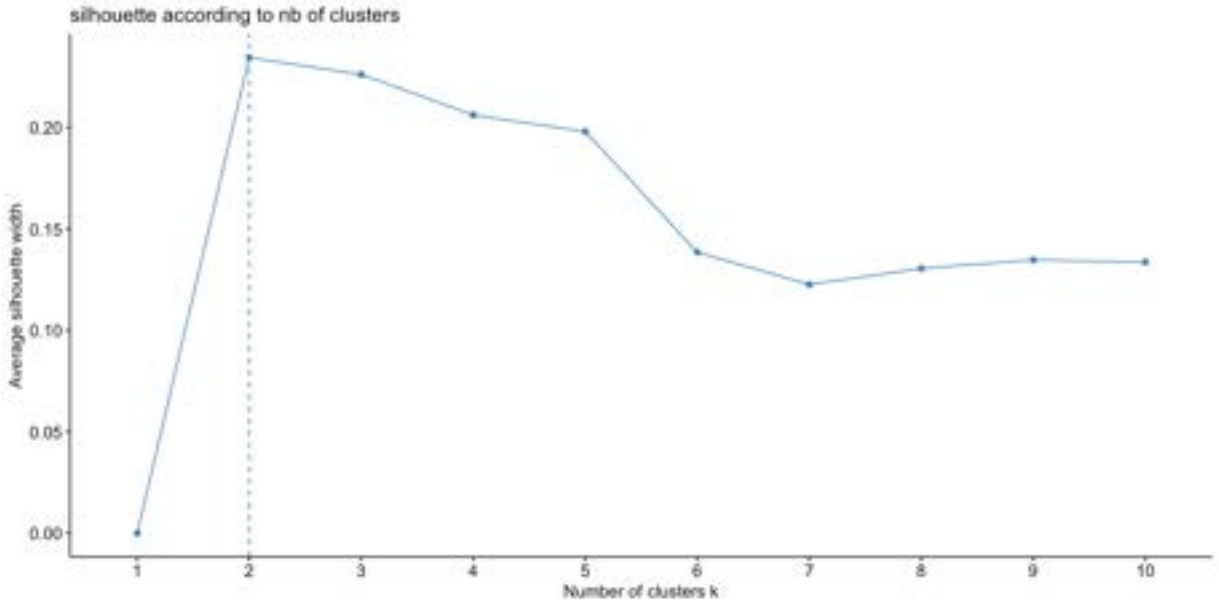
3.2.1.1. Determining the number of clusters using the total within sum of square metric

In [29]:
```
options(repr.plot.width = 12, repr.plot.height = 6)

fviz_nbclust(velib$data[dates], FUNcluster=hcut, method="wss") + ggtitle("WS
```



3.2.1.2. Determining the number of clusters using the silhouette metric

In [30]:
```
fviz_nbclust(velib$data[dates], FUNcluster=hcut, method="silhouette") + ggti
```

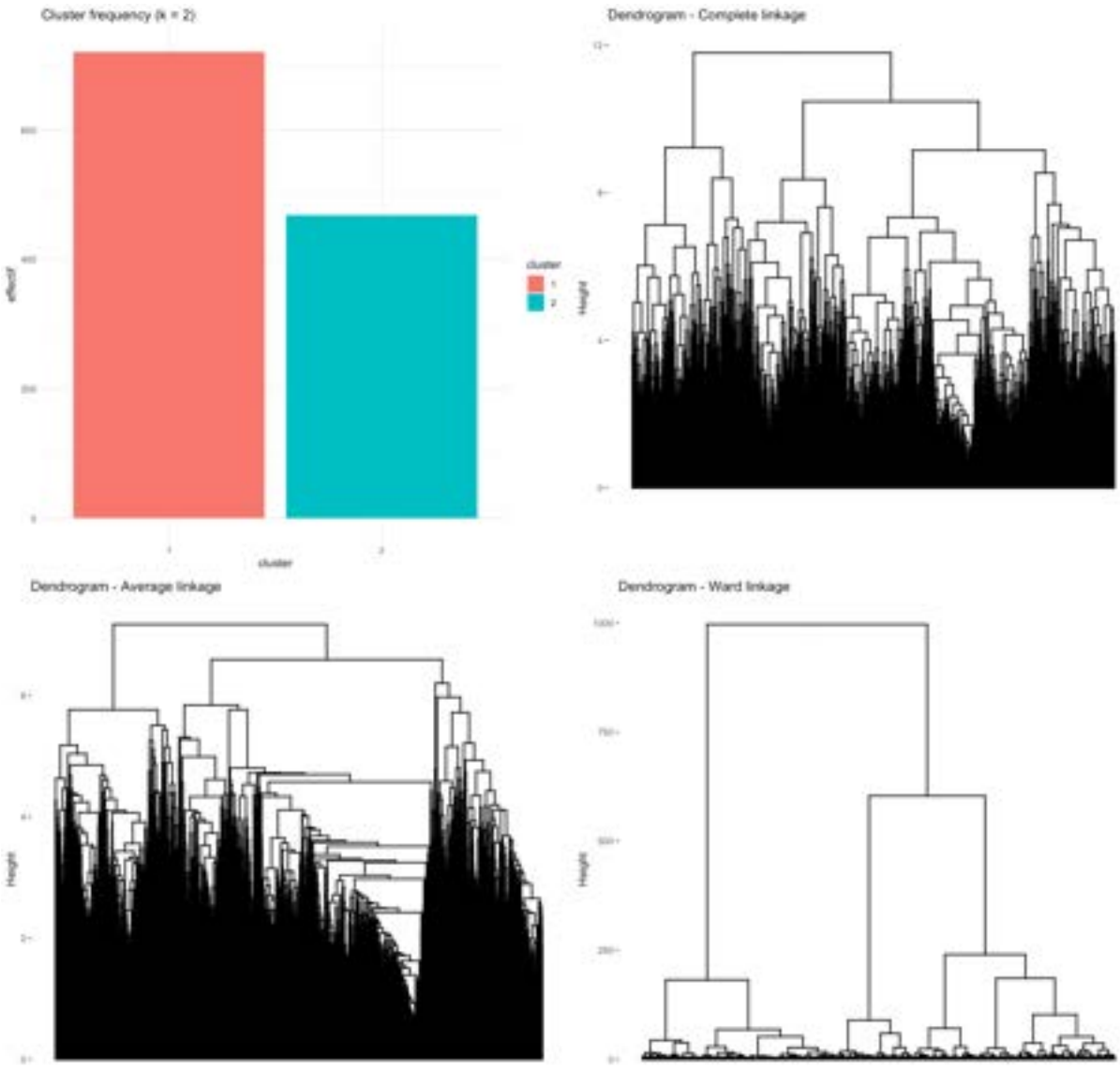


3.2.2. Visualization of different dendrograms and evaluation of the effect of the choice of the linkage function

3.2.2.1. Dendrogram with different linkage methods

In [31]:
```
d = dist(loading, method="euclidean")
options(repr.plot.width=15, repr.plot.height=15)

hclustsingle = hclust(d, method="single")
hclustcomplete = hclust(d, method="complete")
hclustaverage = hclust(d, method="average")
hclustward = hclust(d, method="ward.D")


p1 <- fviz_dend(hclustsingle, show_labels=FALSE, main='Dendrogram - Single l
print('p1 done :)')
p2 <- fviz_dend(hclustcomplete, show_labels=FALSE, main='Dendrogram - Comple
print('p2 done :)')
p3 <- fviz_dend(hclustaverage, show_labels=FALSE, main='Dendrogram - Average
print('p3 done :)')
p4 <- fviz_dend(hclustward, show_labels=FALSE, main='Dendrogram - Ward linka
print('p4 done :)')

#ggpubr::ggarrange(p1,p2,p3,p4)
gridExtra::grid.arrange(p1,p2,p3,p4)
```

```
[1] "p1 done :)"
Warning message:
“The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" instead
as
of ggplot2 3.3.4.
ℹ The deprecated feature was likely used in the factoextra package.
  Please report the issue at <https://github.com/kassambara/factoextra/issue
s>.”
[1] "p2 done :)"
[1] "p3 done :)"
[1] "p4 done :)"
```



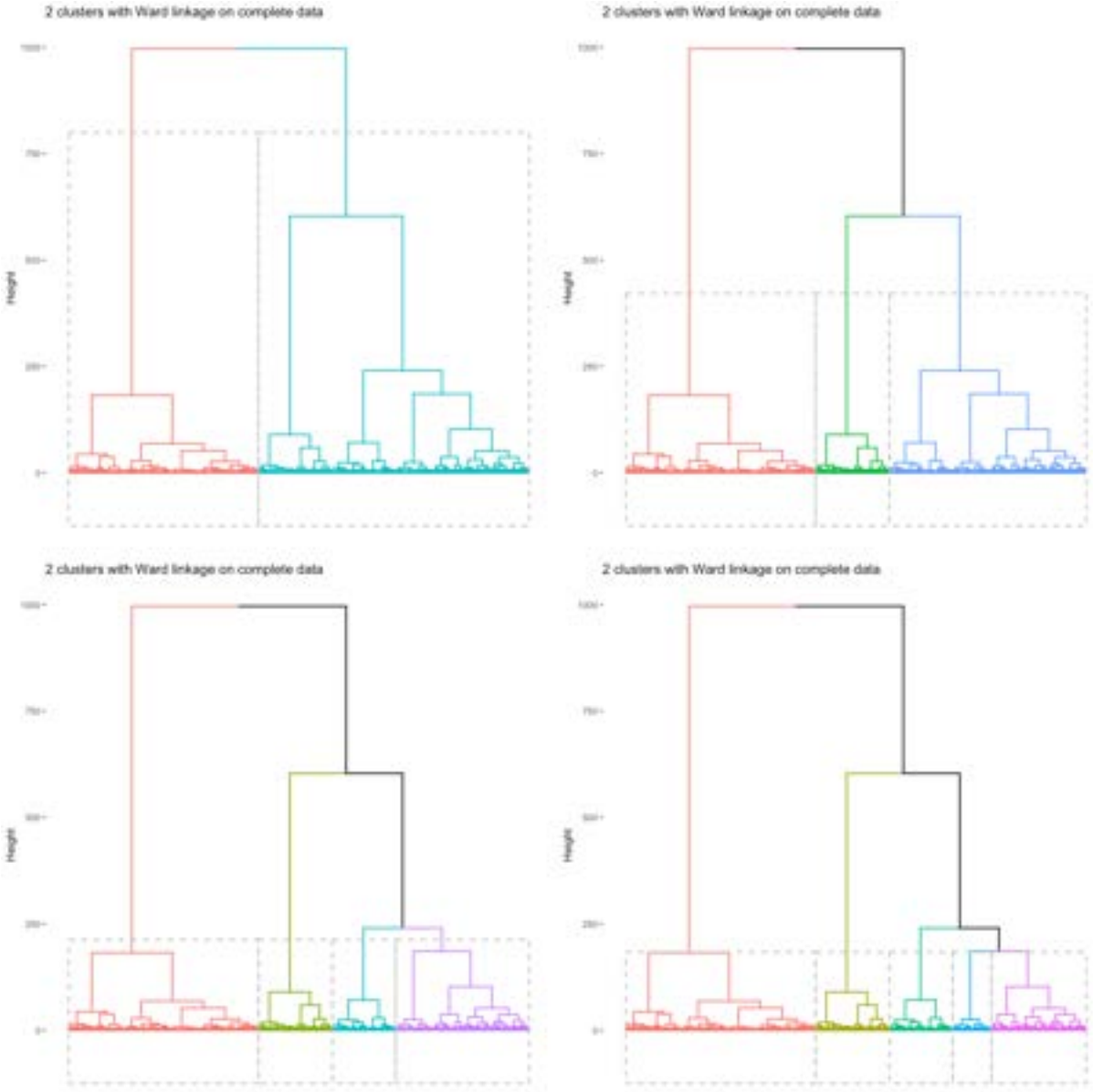
For some unknown reason we were unable to generate the dendrogram with single linkage for the complete data. It did however work with the reduced data (cf. Part 4)

3.2.2.2. Dendrograms with Ward linkage

In [32]:
```
d = dist(loading, method="euclidean")
options(repr.plot.width=15, repr.plot.height=15)

hclustward = hclust(d, method="ward.D")

p1 <- fviz_dend(hclustward, k=2, show_labels=FALSE, rect=TRUE, main='2 clust
print('p1 done :)')
p2 <- fviz_dend(hclustward, k=3, show_labels=FALSE, rect=TRUE, main='2 clust
print('p2 done :)')
p3 <- fviz_dend(hclustward, k=4, show_labels=FALSE, rect=TRUE, main='2 clust
print('p3 done :)')
p4 <- fviz_dend(hclustward, k=5, show_labels=FALSE, rect=TRUE, main='2 clust
print('p4 done :)')
```

```
#ggpubr::ggarrange(p1,p2,p3,p4)
gridExtra::grid.arrange(p1,p2,p3,p4)
```

```
[1] "p1 done :)"
[1] "p2 done :)"
[1] "p3 done :)"
[1] "p4 done :)"
```



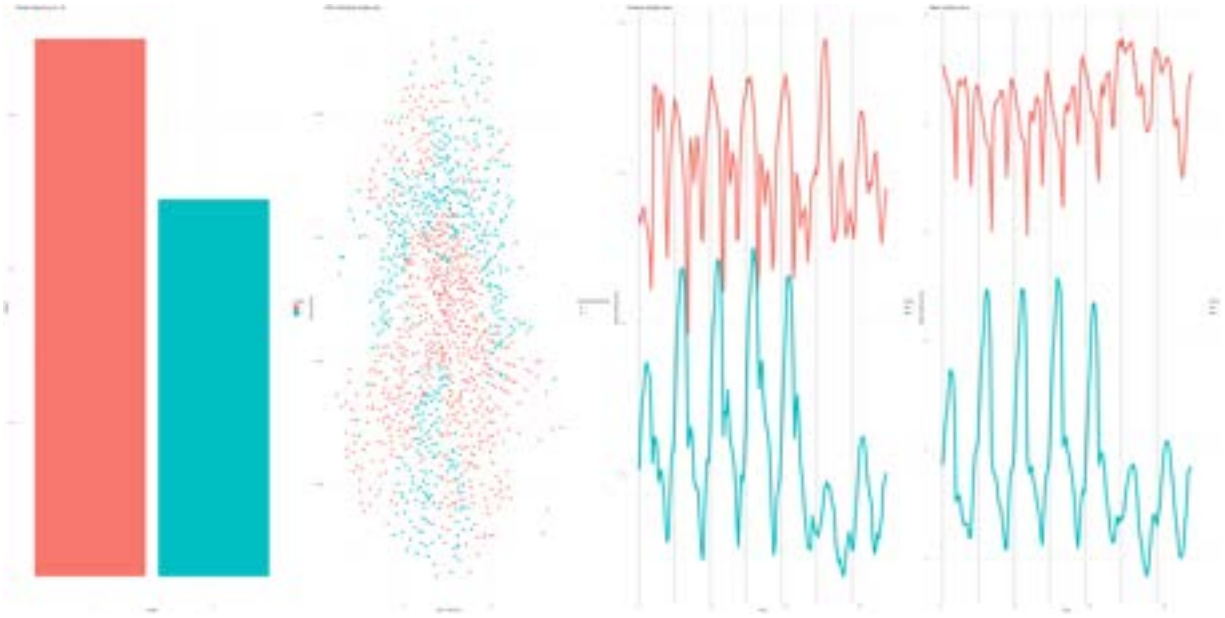
3.2.3. Visualization and interpretation of k-means clusters

3.2.3.1. Four descriptive plots of cluster

In [33]:
```
hclustward2 = cutree(hclustward, k = 2)
hclustward3 = cutree(hclustward, k = 3)
hclustward4 = cutree(hclustward, k = 4)
hclustward5 = cutree(hclustward, k = 5)
```

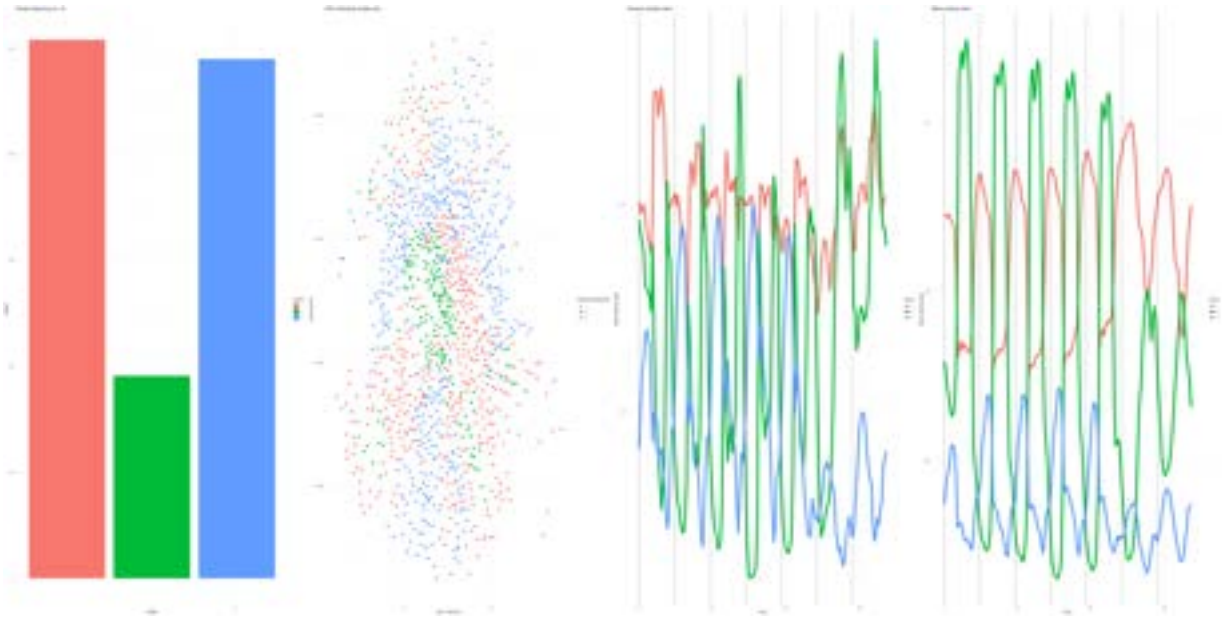
In [34]:
```
time_tick = 1 + 24*(0:6)
```

```
options(repr.plot.width = 50, repr.plot.height = 25)
#size=c(sum(hclustward2==1), sum(hclustward2==2))

###############
#### k = 2 ####
###############

df = data.frame(cluster = c("1", "2"), effectif = c(sum(hclustward2==1), sum

load1 = loading[hclustward2==1,]
load2 = loading[hclustward2==2,]

meanload1 = colMeans(load1)
meanload2 = colMeans(load2)
dfmean = as.data.frame(meanload1)
dfmean$meanload2 = meanload2
time_range = 1:ncol(loading)
dfmean$time_range = time_range

varload1 = colVars(load1)
varload2 = colVars(load2)
dfvar = as.data.frame(varload1)
dfvar$varload2 = varload2
time_range = 1:ncol(loading)
dfvar$time_range = time_range

p1 = ggplot(df, aes(x = cluster, y = effectif, fill=cluster)) +
     geom_bar(stat='identity') +
     labs(title = "Cluster frequency (k = 2)") +
     theme_minimal()

p2 = ggplot(loading, aes(x = coord$longitude, y = coord$latitude)) +
     geom_point(aes(color = factor(hclustward2)), size = 3) +
     labs(title = "HAC individuals scatter plot", x = "Dim 1 (40.5%)", y = "
     theme_minimal()

p3 = ggplot(dfvar,aes(x=time_range)) +
    geom_line(aes(y=varload1, color='1'), linewidth = 3) +
    geom_line(aes(y=varload2, color='2'), linewidth = 3) +
    labs(title = "Variance loading value", x = "Hour", y = "Mean loading val
    geom_vline(xintercept=time_tick, linetype="dashed") +
    theme_minimal()

p4 = ggplot(dfmean,aes(x=time_range)) +
    geom_line(aes(y=meanload1, color='1'), linewidth = 3) +
    geom_line(aes(y=meanload2, color='2'), linewidth = 3) +
    labs(title = "Mean loading value", x = "Hour", y = "Mean loading value")
    geom_vline(xintercept=time_tick, linetype="dashed") +
    theme_minimal()

ggarrange(p1,p2,p3,p4, ncol = 4)
```



In [35]:
```
###############
#### k = 3 ####
###############

df = data.frame(cluster = c("1", "2", "3"), effectif = c(sum(hclustward3==1)

load1 = loading[hclustward3==1,]
load2 = loading[hclustward3==2,]
load3 = loading[hclustward3==3,]

meanload1 = colMeans(load1)
meanload2 = colMeans(load2)
meanload3 = colMeans(load3)
dfmean = as.data.frame(meanload1)
dfmean$meanload2 = meanload2
dfmean$meanload3 = meanload3
time_range = 1:ncol(loading)
dfmean$time_range = time_range

varload1 = colVars(load1)
varload2 = colVars(load2)
varload3 = colVars(load3)
dfvar = as.data.frame(varload1)
dfvar$varload2 = varload2
dfvar$varload3 = varload3
time_range = 1:ncol(loading)
dfvar$time_range = time_range

p1 = ggplot(df, aes(x = cluster, y = effectif, fill=cluster)) +
     geom_bar(stat='identity') +
     labs(title = "Cluster frequency (k = 3)") +
     theme_minimal()

p2 = ggplot(loading, aes(x = coord$longitude, y = coord$latitude)) +
    geom_point(aes(color = factor(hclustward3)), size = 3) +
    labs(title = "HAC individuals scatter plot", x = "Dim 1 (40.5%)", y = "D
    theme_minimal()
```

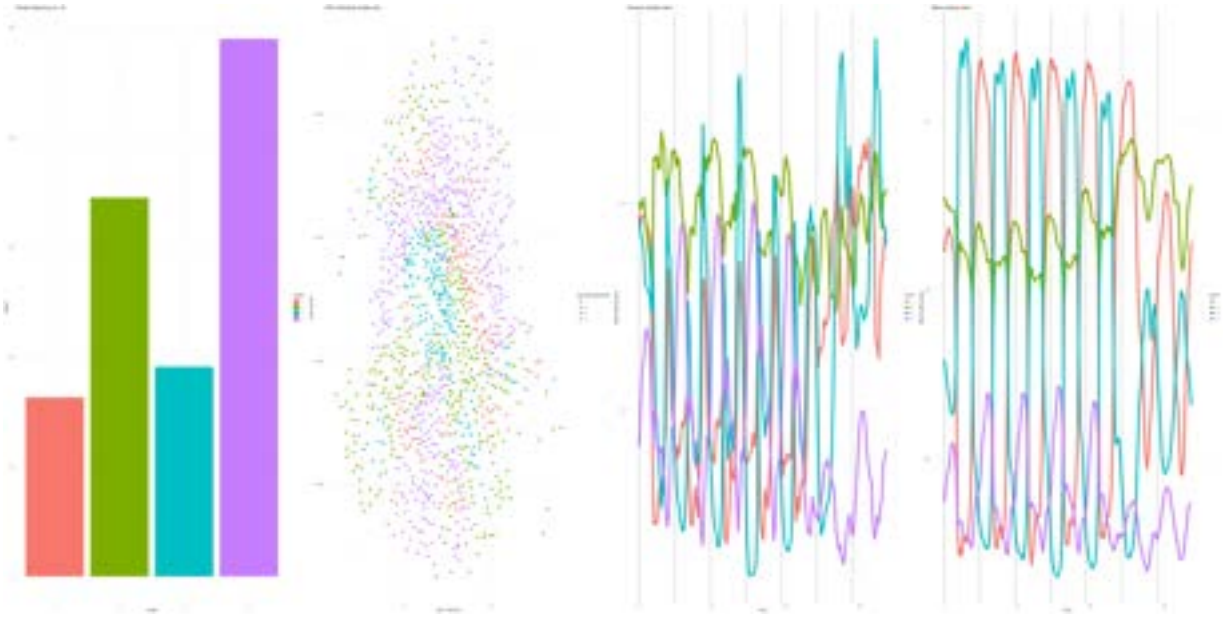
```
p3 = ggplot(dfvar,aes(x=time_range)) +
    geom_line(aes(y=varload1, color='1'), linewidth = 3) +
    geom_line(aes(y=varload2, color='2'), linewidth = 3) +
    geom_line(aes(y=varload3, color='3'), linewidth = 3) +
    labs(title = "Variance loading value", x = "Hour", y = "Mean loading val
    geom_vline(xintercept=time_tick, linetype="dashed") +
    theme_minimal()

p4 = ggplot(dfmean,aes(x=time_range)) +
    geom_line(aes(y=meanload1, color='1'), linewidth = 3) +
    geom_line(aes(y=meanload2, color='2'), linewidth = 3) +
    geom_line(aes(y=meanload3, color='3'), linewidth = 3) +
    labs(title = "Mean loading value", x = "Hour", y = "Mean loading value")
    geom_vline(xintercept=time_tick, linetype="dashed") +
    theme_minimal()

ggarrange(p1,p2,p3,p4, ncol = 4)
```



In [36]:

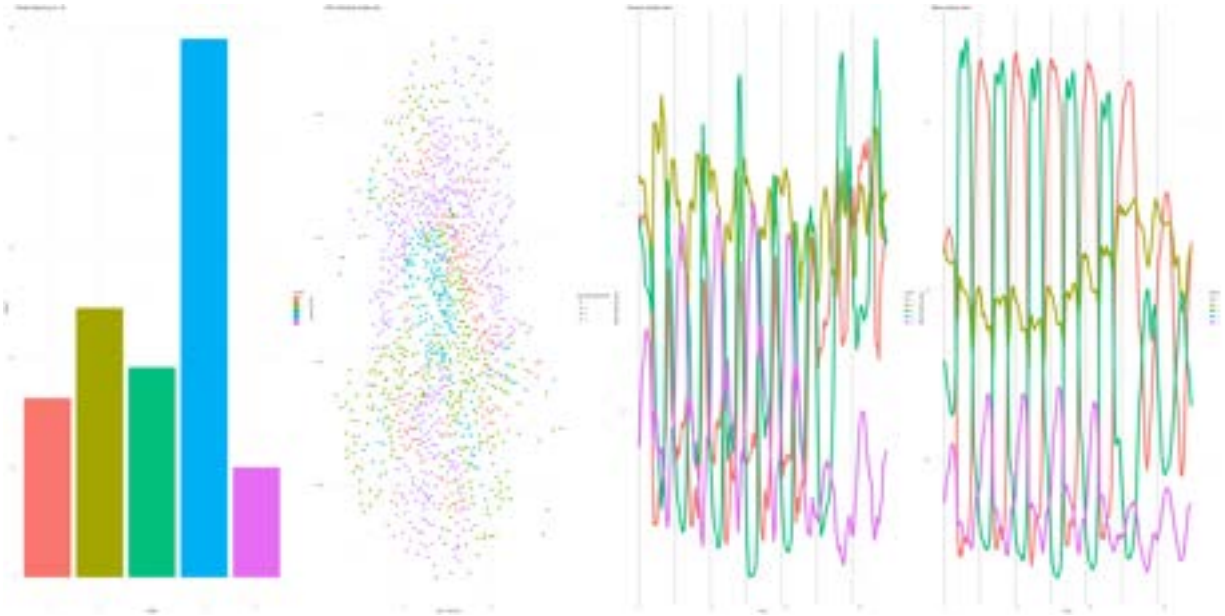
```
###############
#### k = 4 ####
###############

df = data.frame(cluster = c("1", "2", "3", "4"), effectif = c(sum(hclustward

load1 = loading[hclustward4==1,]
load2 = loading[hclustward4==2,]
load3 = loading[hclustward4==3,]
load4 = loading[hclustward4==4,]

meanload1 = colMeans(load1)
meanload2 = colMeans(load2)
meanload3 = colMeans(load3)
meanload4 = colMeans(load4)
dfmean = as.data.frame(meanload1)
dfmean$meanload2 = meanload2
dfmean$meanload3 = meanload3
```

```
dfmean$meanload4 = meanload4
time_range = 1:ncol(loading)
dfmean$time_range = time_range

varload1 = colVars(load1)
varload2 = colVars(load2)
varload3 = colVars(load3)
varload4 = colVars(load4)
dfvar = as.data.frame(varload1)
dfvar$varload2 = varload2
dfvar$varload3 = varload3
dfvar$varload4 = varload4
time_range = 1:ncol(loading)
dfvar$time_range = time_range

p1 = ggplot(df, aes(x = cluster, y = effectif, fill=cluster)) +
     geom_bar(stat='identity') +
     labs(title = "Cluster frequency (k = 4)") +
     theme_minimal()

p2 = ggplot(loading, aes(x = coord$longitude, y = coord$latitude)) +
    geom_point(aes(color = factor(hclustward4)), size=3) +
    labs(title = "HAC individuals scatter plot", x = "Dim 1 (40.5%)", y = "D
    theme_minimal()

p3 = ggplot(dfvar,aes(x=time_range)) +
    geom_line(aes(y=varload1, color='1'), linewidth = 3) +
    geom_line(aes(y=varload2, color='2'), linewidth = 3) +
    geom_line(aes(y=varload3, color='3'), linewidth = 3) +
    geom_line(aes(y=varload4, color='4'), linewidth = 3) +
    labs(title = "Variance loading value", x = "Hour", y = "Mean loading val
    geom_vline(xintercept=time_tick, linetype="dashed") +
    theme_minimal()

p4 = ggplot(dfmean,aes(x=time_range)) +
    geom_line(aes(y=meanload1, color='1'), linewidth = 3) +
    geom_line(aes(y=meanload2, color='2'), linewidth = 3) +
    geom_line(aes(y=meanload3, color='3'), linewidth = 3) +
    geom_line(aes(y=meanload4, color='4'), linewidth = 3) +
    labs(title = "Mean loading value", x = "Hour", y = "Mean loading value")
    geom_vline(xintercept=time_tick, linetype="dashed") +
    theme_minimal()

ggarrange(p1,p2,p3,p4, ncol = 4)
```



In [37]:
```
###############
#### k = 5 ####
###############

df = data.frame(cluster = c("1", "2", "3", "4", "5"), effectif = c(sum(hclus

load1 = loading[hclustward5==1,]
load2 = loading[hclustward5==2,]
load3 = loading[hclustward5==3,]
load4 = loading[hclustward5==4,]
load5 = loading[hclustward5==5,]

meanload1 = colMeans(load1)
meanload2 = colMeans(load2)
meanload3 = colMeans(load3)
meanload4 = colMeans(load4)
meanload5 = colMeans(load5)
dfmean = as.data.frame(meanload1)
dfmean$meanload2 = meanload2
dfmean$meanload3 = meanload3
dfmean$meanload4 = meanload4
dfmean$meanload5 = meanload5
time_range = 1:ncol(loading)
dfmean$time_range = time_range

varload1 = colVars(load1)
varload2 = colVars(load2)
varload3 = colVars(load3)
varload4 = colVars(load4)
varload5 = colVars(load5)
dfvar = as.data.frame(varload1)
dfvar$varload2 = varload2
dfvar$varload3 = varload3
dfvar$varload4 = varload4
dfvar$varload5 = varload5
time_range = 1:ncol(loading)
dfvar$time_range = time_range
```

```
p1 = ggplot(df, aes(x = cluster, y = effectif, fill=cluster)) +
     geom_bar(stat='identity') +
     labs(title = "Cluster frequency (k = 5)") +
     theme_minimal()

p2 = ggplot(loading, aes(x = coord$longitude, y = coord$latitude)) +
    geom_point(aes(color = factor(hclustward4)), size=3) +
    labs(title = "HAC individuals scatter plot", x = "Dim 1 (40.5%)", y = "D
    theme_minimal()

p3 = ggplot(dfvar,aes(x=time_range)) +
    geom_line(aes(y=varload1, color='1'), linewidth = 3) +
    geom_line(aes(y=varload2, color='2'), linewidth = 3) +
    geom_line(aes(y=varload3, color='3'), linewidth = 3) +
    geom_line(aes(y=varload4, color='4'), linewidth = 3) +
    geom_line(aes(y=varload4, color='5'), linewidth = 3) +
    labs(title = "Variance loading value", x = "Hour", y = "Mean loading val
    geom_vline(xintercept=time_tick, linetype="dashed") +
    theme_minimal()

p4 = ggplot(dfmean,aes(x=time_range)) +
    geom_line(aes(y=meanload1, color='1'), linewidth = 3) +
    geom_line(aes(y=meanload2, color='2'), linewidth = 3) +
    geom_line(aes(y=meanload3, color='3'), linewidth = 3) +
    geom_line(aes(y=meanload4, color='4'), linewidth = 3) +
    geom_line(aes(y=meanload4, color='5'), linewidth = 3) +
    labs(title = "Mean loading value", x = "Hour", y = "Mean loading value")
    geom_vline(xintercept=time_tick, linetype="dashed") +
    theme_minimal()

ggarrange(p1,p2,p3,p4, ncol = 4)
```

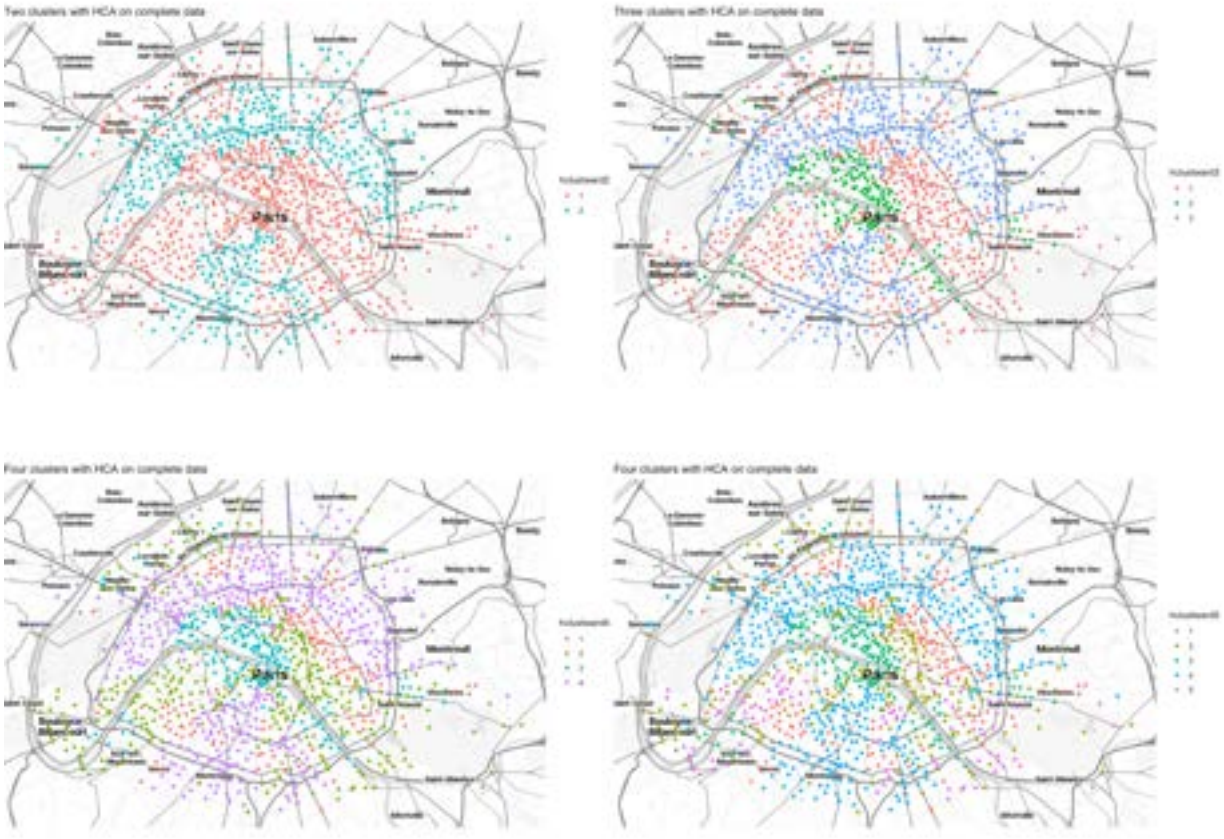


3.2.3.2. Visualization of clusters on the map

In [38]:
```
options(repr.plot.width = 20, repr.plot.height = 15)
```

```
##############
hclustward2 = as.factor(hclustward2)

df2 = data.frame(size=c(sum(hclustward2==1), sum(hclustward2==2)),
                 labels = c('cluster 1','cluster 2'))

p2 = qmplot(data=coord, longitude, latitude, color=hclustward2) +
    #scale_color_manual(values = c("1" = green, "2" = orange)) +
    labs(title = 'Two clusters with HCA on complete data')

##############
hclustward3 = as.factor(hclustward3)

df3 = data.frame(size=c(sum(hclustward3==1), sum(hclustward3==2), sum(hclust
                 labels = c('cluster 1','cluster 2', 'cluster 3'))

p3 = qmplot(data=coord, longitude, latitude, color=hclustward3) +
    #scale_color_manual(values = c("1" = green, "2" = orange, "3" = purple))
    labs(title = 'Three clusters with HCA on complete data')

##############
hclustward4 = as.factor(hclustward4)

df4 = data.frame(size=c(sum(hclustward4==1), sum(hclustward4==2), sum(hclust
                 labels = c('cluster 1','cluster 2', 'cluster 3', 'cluster 4'

p4 = qmplot(data=coord, longitude, latitude, color=hclustward4) +
    #scale_color_manual(values = c("1" = green, "2" = orange, "3" = purple,
    labs(title = 'Four clusters with HCA on complete data')

##############
hclustward5 = as.factor(hclustward5)

df5 = data.frame(size=c(sum(hclustward5==1), sum(hclustward5==2), sum(hclust
                 labels = c('cluster 1','cluster 2', 'cluster 3', 'cluster 4'

p5 = qmplot(data=coord, longitude, latitude, color=hclustward5) +
    #scale_color_manual(values = c("1" = green, "2" = orange, "3" = purple,
    labs(title = 'Four clusters with HCA on complete data')

ggpubr::ggarrange(p2, p3, p4, p5)
```

```
ℹ Using `zoom = 12`
ℹ © Stadia Maps © Stamen Design © OpenMapTiles © OpenStreetMap contributors.
ℹ Using `zoom = 12`
ℹ © Stadia Maps © Stamen Design © OpenMapTiles © OpenStreetMap contributors.
ℹ Using `zoom = 12`
ℹ © Stadia Maps © Stamen Design © OpenMapTiles © OpenStreetMap contributors.
ℹ Using `zoom = 12`
ℹ © Stadia Maps © Stamen Design © OpenMapTiles © OpenStreetMap contributors.
```



3.3. Gaussian Mixture clustering on original data

3.3.1. Selection of the number of clusters

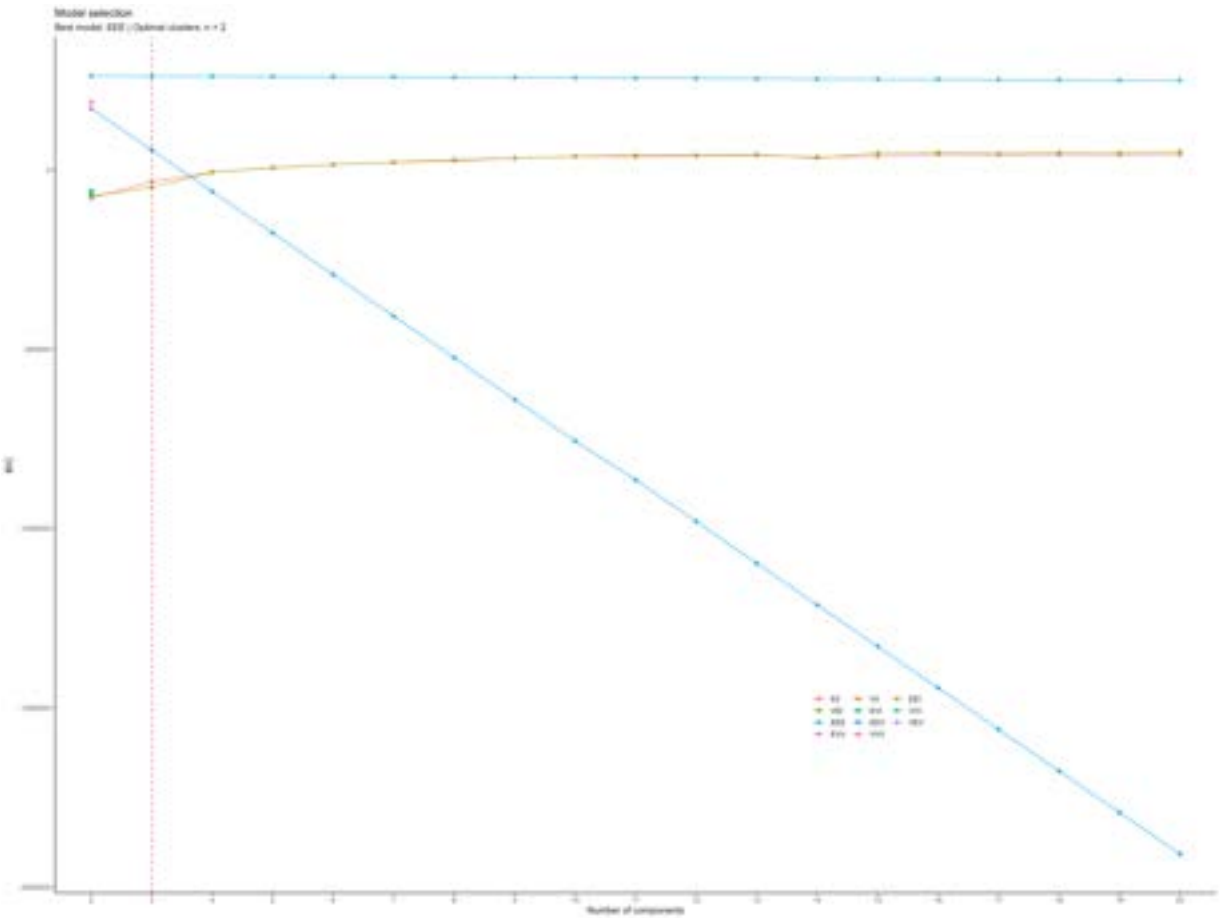
3.3.1.1. Determining the number of clusters using BIC

In [39]:
```
resBICall = mclustBIC(velib$data, G=2:20)
summary(resBICall)

resBICall = Mclust(velib$data, G=2:20)
```

```
summary(resBICall)

fviz_mclust(resBICall, what="BIC")
```

```
Best BIC values:
          EEE,2       EEE,3       EEE,4
BIC      261311 260888.3422 260462.8096
BIC diff      0   -422.6327   -848.1654
----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------

Mclust EEE (ellipsoidal, equal volume, shape and orientation) model with 2
components:

 log-likelihood    n    df    BIC      ICL
       190255.2 1189 16834 261311 261295.4

Clustering table:
  1   2
665 524
```

```
Warning message:
“`gather_()` was deprecated in tidyr 1.2.0.
ℹ Please use `gather()` instead.
ℹ The deprecated feature was likely used in the factoextra package.
  Please report the issue at <https://github.com/kassambara/factoextra/issue
s>.”
```



3.3.1.2. Determining the number of clusters using ICL

In [40]:
```
resICLall = mclustICL(velib$data, G=2:20)
summary(resICLall)
```

```
Best ICL values:
            EEE,2       EEE,3       EEE,4
ICL      261295.4 260869.5215 260443.5140
ICL diff      0.0   -425.9087   -851.9162
```

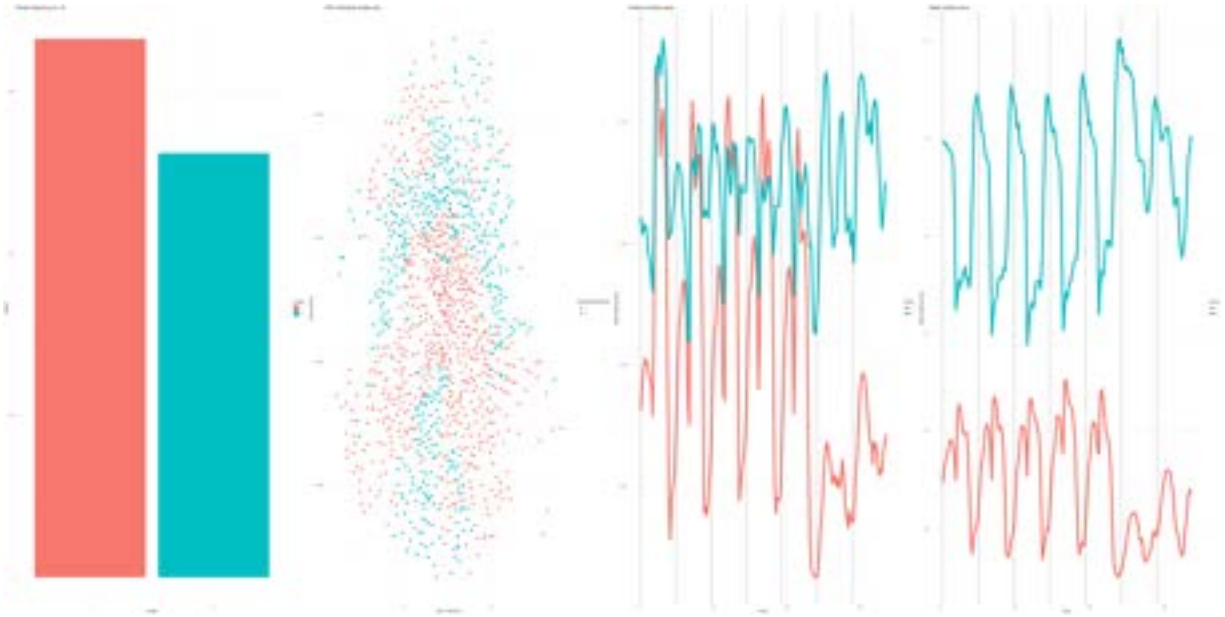
Both methods say that the best model is EEE with 2 clusters

3.3.2. Visualization and interpretation of GMM

3.3.2.1 Four descriptive plots of cluster

In [41]:
```
time_tick = 1 + 24*(0:6)
options(repr.plot.width = 50, repr.plot.height = 25)
#size=c(sum(hclustward2==1), sum(hclustward2==2))

###############
#### k = 2 ####
###############

df = data.frame(cluster = c("1", "2"), effectif = c(sum(resBICall$classifica

load1 = loading[resBICall$classification==1,]
load2 = loading[resBICall$classification==2,]

meanload1 = colMeans(load1)
meanload2 = colMeans(load2)
dfmean = as.data.frame(meanload1)
dfmean$meanload2 = meanload2
time_range = 1:ncol(loading)
dfmean$time_range = time_range

varload1 = colVars(load1)
varload2 = colVars(load2)
dfvar = as.data.frame(varload1)
dfvar$varload2 = varload2
time_range = 1:ncol(loading)
dfvar$time_range = time_range

p1 = ggplot(df, aes(x = cluster, y = effectif, fill=cluster)) +
     geom_bar(stat='identity') +
     labs(title = "Cluster frequency (k = 2)") +
     theme_minimal()

p2 = ggplot(loading, aes(x = coord$longitude, y = coord$latitude)) +
     geom_point(aes(color = factor(hclustward2)), size = 3) +
     labs(title = "HAC individuals scatter plot", x = "Dim 1 (40.5%)", y = "
     theme_minimal()

p3 = ggplot(dfvar,aes(x=time_range)) +
```

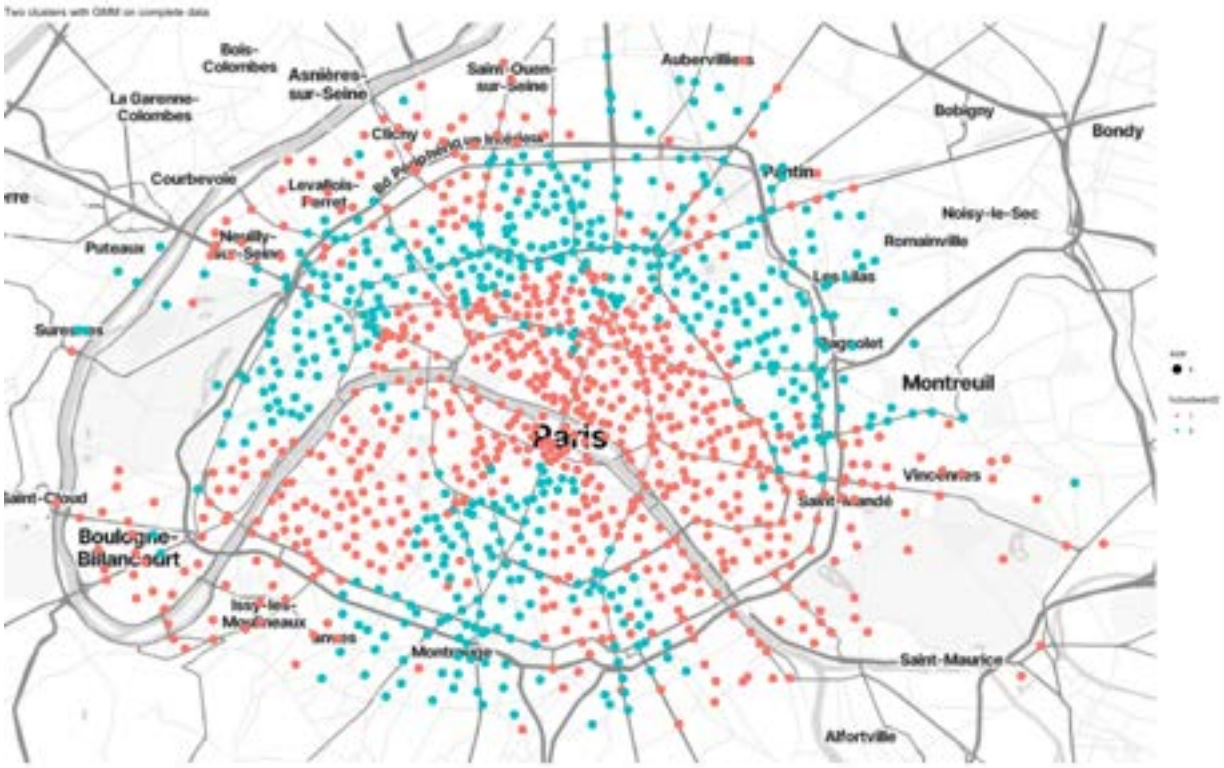
```
    geom_line(aes(y=varload1, color='1'), linewidth = 3) +
    geom_line(aes(y=varload2, color='2'), linewidth = 3) +
    labs(title = "Variance loading value", x = "Hour", y = "Mean loading val
    geom_vline(xintercept=time_tick, linetype="dashed") +
    theme_minimal()

p4 = ggplot(dfmean,aes(x=time_range)) +
    geom_line(aes(y=meanload1, color='1'), linewidth = 3) +
    geom_line(aes(y=meanload2, color='2'), linewidth = 3) +
    labs(title = "Mean loading value", x = "Hour", y = "Mean loading value")
    geom_vline(xintercept=time_tick, linetype="dashed") +
    theme_minimal()

ggarrange(p1,p2,p3,p4, ncol = 4)
```



3.3.2.2. Visualization of clusters on the map

In [42]:
```
options(repr.plot.width = 20, repr.plot.height = 15)

###############
resBICall$classification = as.factor(resBICall$classification)

df2 = data.frame(size=c(sum(resBICall$classification==1), sum(resBICall$clas
                labels = c('cluster 1','cluster 2'))

qmplot(data=coord, longitude, latitude, color=hclustward2, size = 5) +
    #scale_color_manual(values = c("1" = green, "2" = orange)) +
    labs(title = 'Two clusters with GMM on complete data')
```

```
ℹ Using `zoom = 12`

ℹ © Stadia Maps © Stamen Design © OpenMapTiles © OpenStreetMap contributors.
```

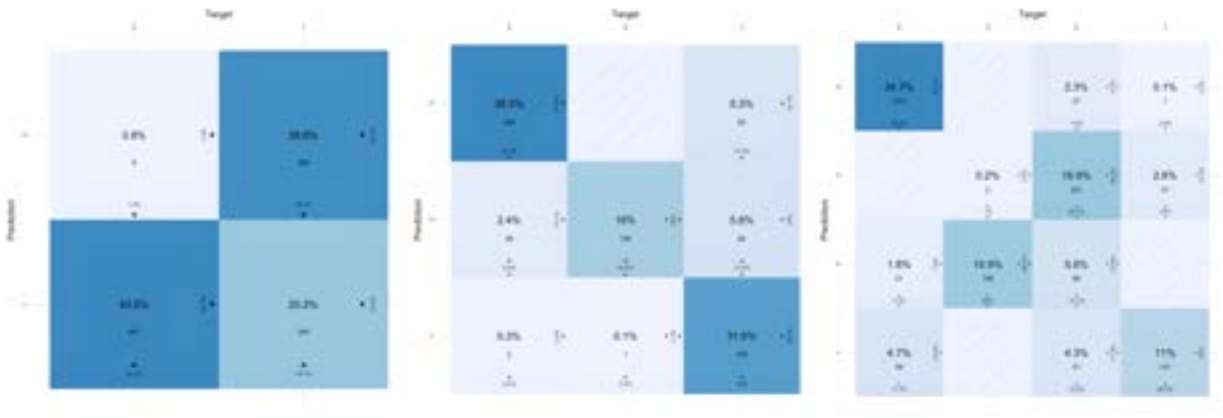


The results are not very good because there are too many parameters to estimate.

3.4. Comparison of clustering methods

3.4.1. HCA vs. k-means

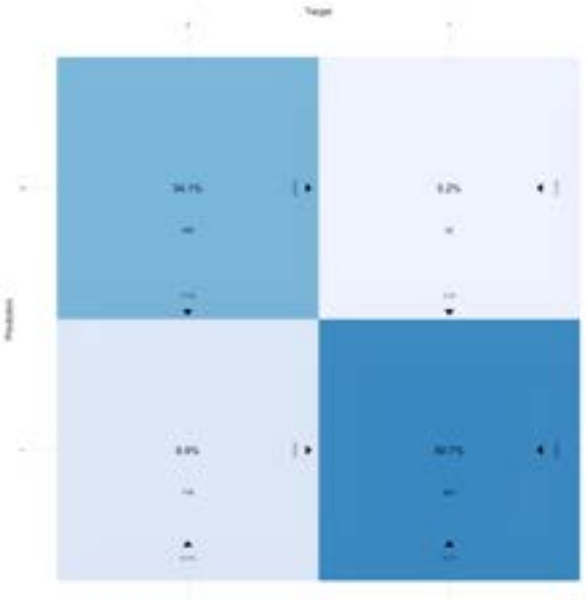
In [43]:
```
options(repr.plot.width=20, repr.plot.height=10)

conf_mat2 = confusion_matrix(targets=hclustward2, predictions=reskmeans2$clu
conf_mat3 = confusion_matrix(targets=hclustward3, predictions=reskmeans3$clu
conf_mat4 = confusion_matrix(targets=hclustward4, predictions=reskmeans4$clu

p1 = plot_confusion_matrix(conf_mat2)
p2 = plot_confusion_matrix(conf_mat3)
p3 = plot_confusion_matrix(conf_mat4)

ggarrange(p1, p2, p3, ncol = 3)
```



Note : we were unable to maximise the diagonal.

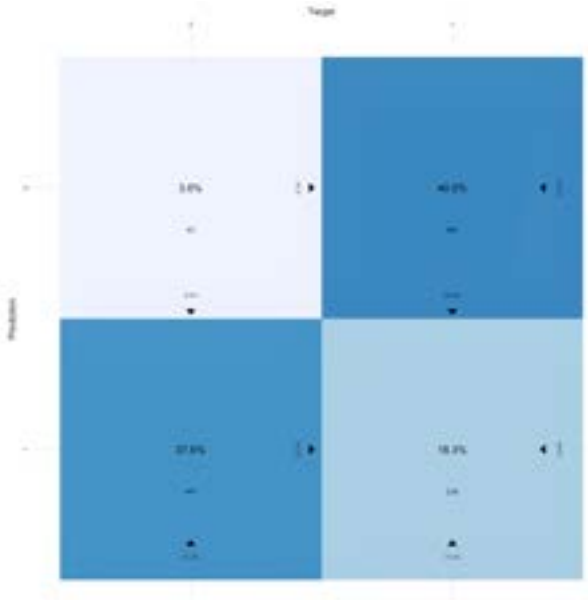
3.4.2. GMM vs. k-means

In [44]:
```
conf_mat = confusion_matrix(targets=resBICall$classification, predictions=re
plot_confusion_matrix(conf_mat)
```



3.4.3. HCA versus GMM

In [45]:
```
conf_mat = confusion_matrix(targets=hclustward2, predictions=resBICall$class
plot_confusion_matrix(conf_mat)
```



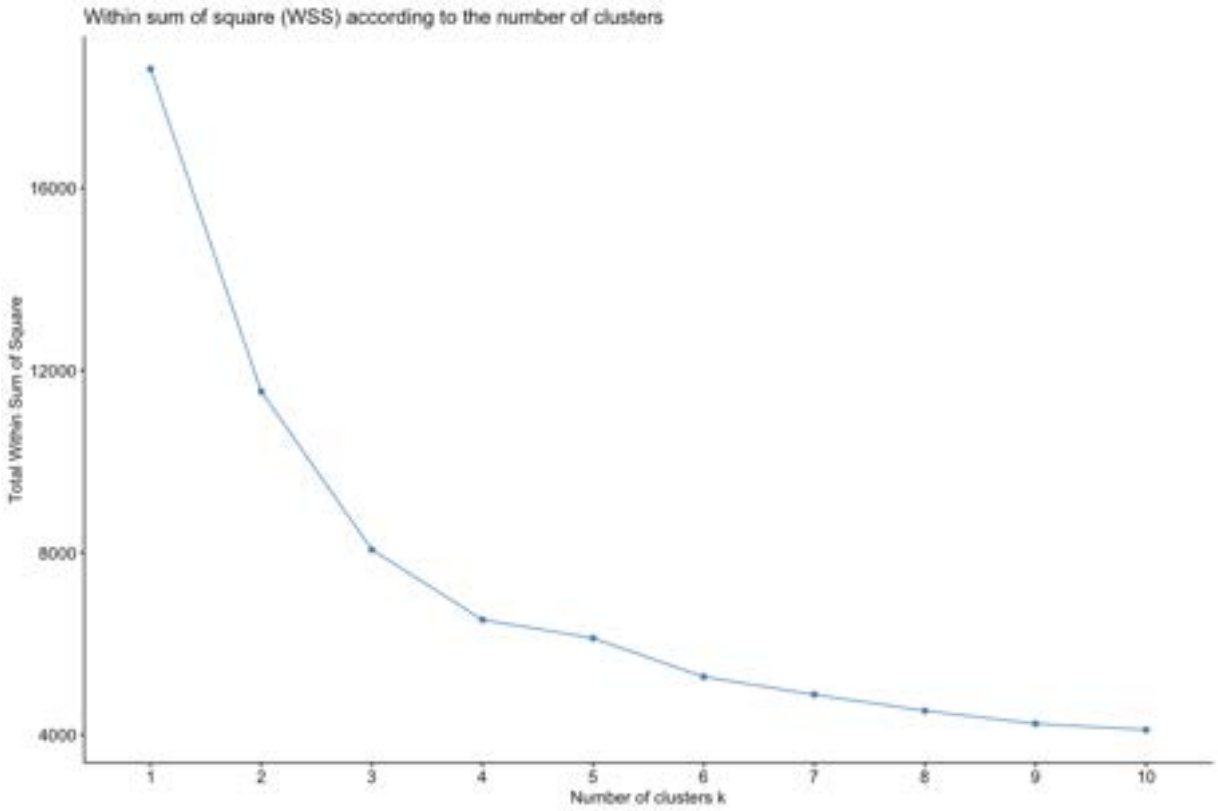
Part 4: clustering on reduced data

4.1. K-means clustering

4.1.1. Selection of the number of clusters

4.1.1.1. Determining the number of clusters using the total within sum of square metric

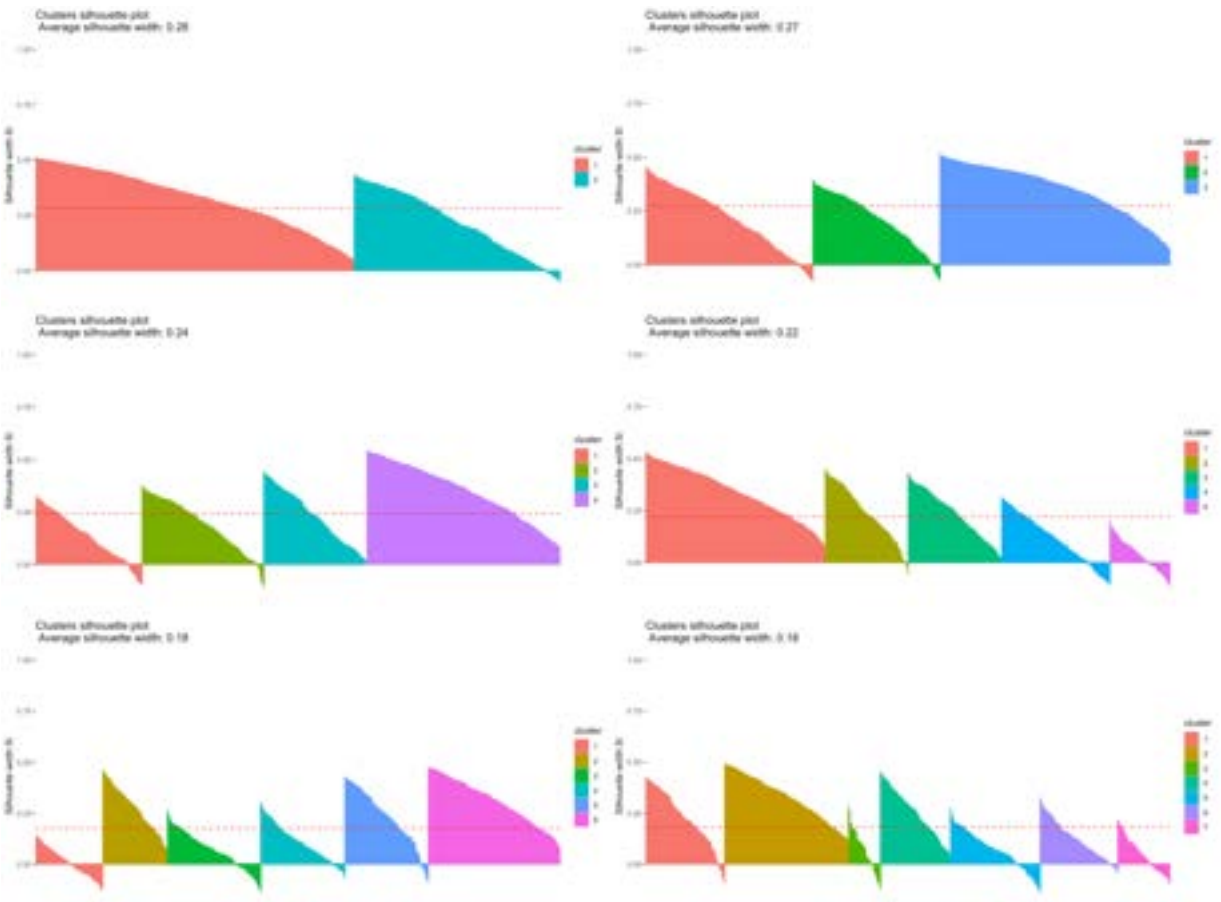
In [46]:
```
options(repr.plot.width = 12, repr.plot.height = 8)

fviz_nbclust(pca$ind$coord, FUNcluster=kmeans, method="wss") +
    ggtitle("Within sum of square (WSS) according to the number of clusters"
```



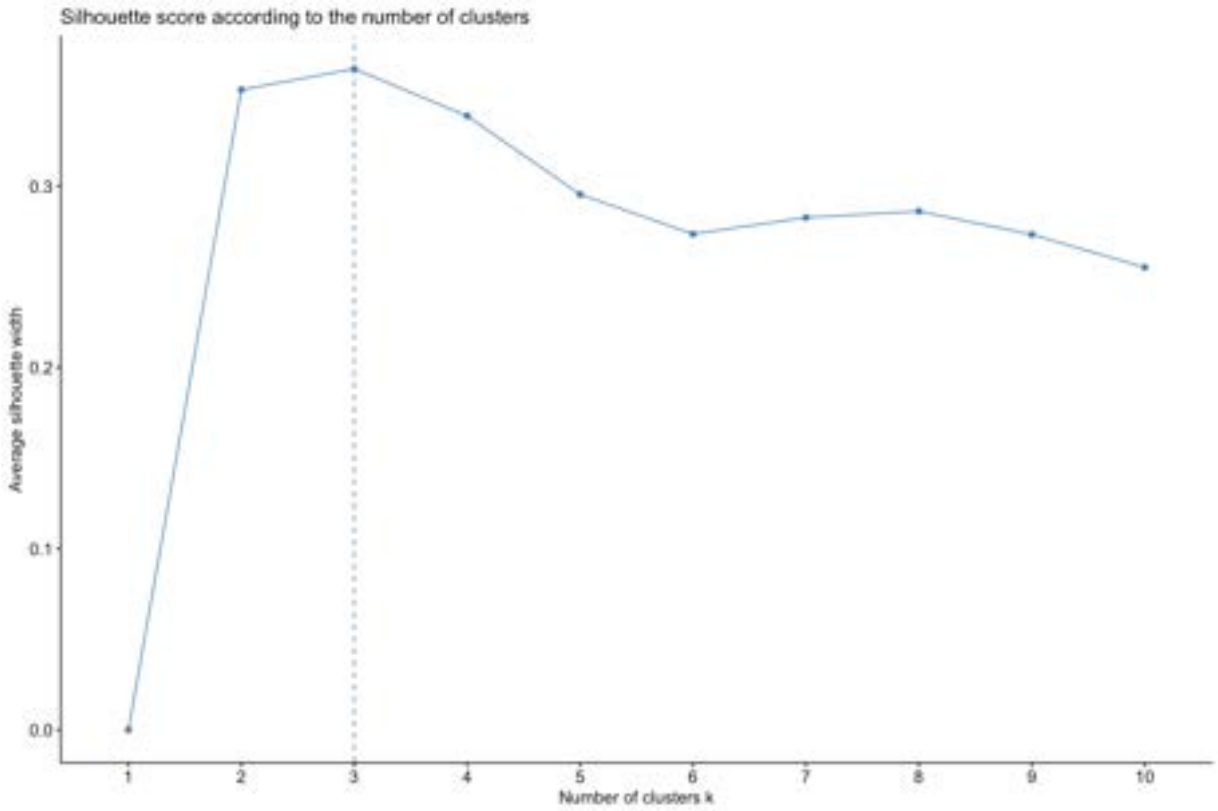
4.1.1.2. Determining the number of clusters using the silhouette scores metric

In [47]:
```
# Silhouette plots, according to the number of clusters
options(repr.plot.width = 20, repr.plot.height = 15)

reskmeans2 = kmeans(pca$ind$coord, centers=2)
reskmeans3 = kmeans(pca$ind$coord, centers=3)
reskmeans4 = kmeans(pca$ind$coord, centers=4)
reskmeans5 = kmeans(pca$ind$coord, centers=5)
reskmeans6 = kmeans(pca$ind$coord, centers=6)
reskmeans7 = kmeans(pca$ind$coord, centers=7)


sil = silhouette(reskmeans2$cluster, dist(loading))
p1 = fviz_silhouette(sil, print.summary = FALSE)

sil = silhouette(reskmeans3$cluster, dist(loading))
p2 = fviz_silhouette(sil, print.summary = FALSE)

sil = silhouette(reskmeans4$cluster, dist(loading))
p3 = fviz_silhouette(sil, print.summary = FALSE)

sil = silhouette(reskmeans5$cluster, dist(loading))
p4 = fviz_silhouette(sil, print.summary = FALSE)

sil = silhouette(reskmeans6$cluster, dist(loading))
p5 = fviz_silhouette(sil, print.summary = FALSE)

sil = silhouette(reskmeans7$cluster, dist(loading))
```

```
p6 = fviz_silhouette(sil, print.summary = FALSE)

grid.arrange(p1,p2,p3,p4,p5,p6, ncol=2)
```



In [48]:
```
options(repr.plot.width = 12, repr.plot.height = 8)

fviz_nbclust(pca$ind$coord, FUNcluster=kmeans, method="silhouette") +
    ggtitle("Silhouette score according to the number of clusters")
```



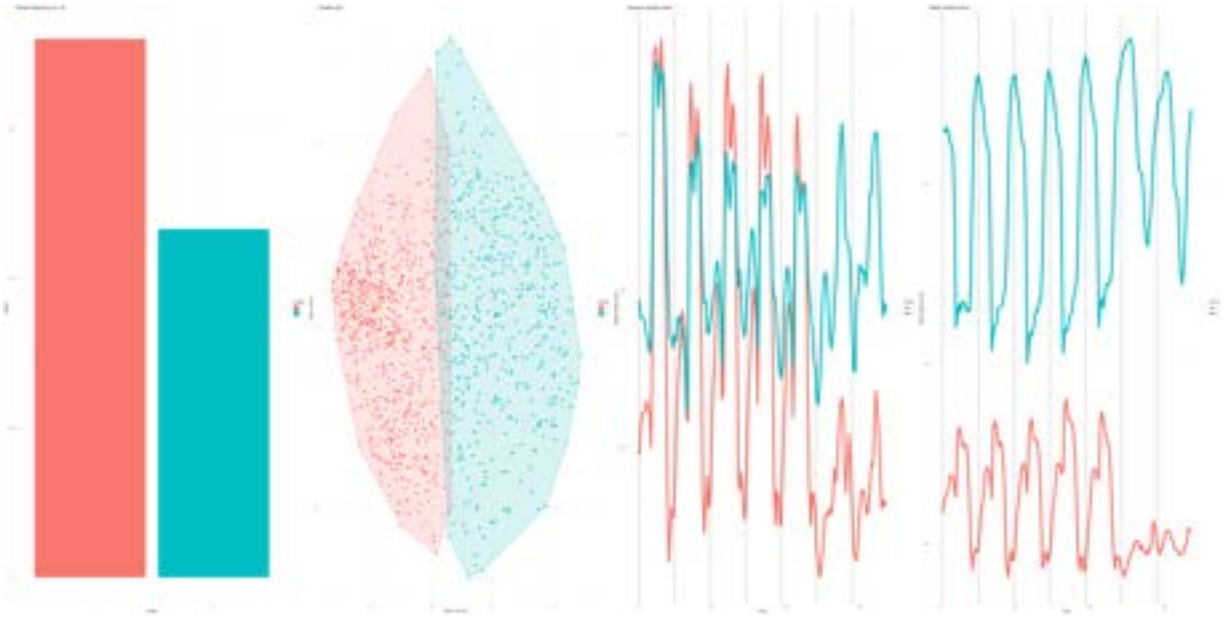
We will perform the study with k in (2, 3, 4)

4.1.2. Visualization and interpretation of k-means clusters

4.1.2.1. Four descriptive plots of cluster

In [49]:
```
time_tick = 1 + 24*(0:6)
options(repr.plot.width = 50, repr.plot.height = 25)

###############
#### k = 2 ####
###############

df = data.frame(cluster = c("1", "2"), effectif = c(reskmeans2$size))

load1 = loading[reskmeans2$cluster==1,]
load2 = loading[reskmeans2$cluster==2,]

meanload1 = colMeans(load1)
meanload2 = colMeans(load2)
dfmean = as.data.frame(meanload1)
dfmean$meanload2 = meanload2
time_range = 1:ncol(loading)
dfmean$time_range = time_range

varload1 = colVars(load1)
varload2 = colVars(load2)
dfvar = as.data.frame(varload1)
dfvar$varload2 = varload2
time_range = 1:ncol(loading)
```

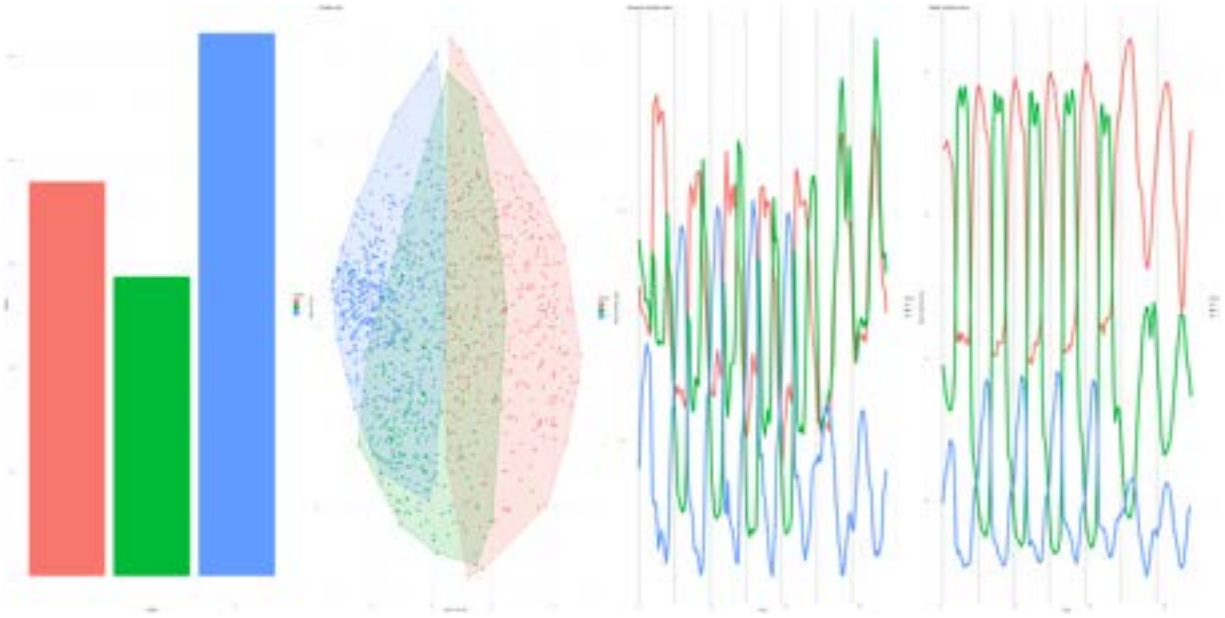
```
dfvar$time_range = time_range

p1 = ggplot(df, aes(x = cluster, y = effectif, fill=cluster)) +
    geom_bar(stat='identity') +
    labs(title = "Cluster frequency (k = 2)") +
    theme_minimal()

p2 = fviz_cluster(reskmeans2, data=pca$ind$coord, ellipse.type="convex", lab

p3 = ggplot(dfvar,aes(x=time_range)) +
    geom_line(aes(y=varload1, color='1'), linewidth = 3) +
    geom_line(aes(y=varload2, color='2'), linewidth = 3) +
    labs(title = "Variance loading value", x = "Hour", y = "Mean loading val
    geom_vline(xintercept=time_tick, linetype="dashed") +
    theme_minimal()

p4 = ggplot(dfmean,aes(x=time_range)) +
    geom_line(aes(y=meanload1, color='1'), linewidth = 3) +
    geom_line(aes(y=meanload2, color='2'), linewidth = 3) +
    labs(title = "Mean loading value", x = "Hour", y = "Mean loading value")
    geom_vline(xintercept=time_tick, linetype="dashed") +
    theme_minimal()

ggarrange(p1,p2,p3,p4, ncol = 4)
```



In [50]:

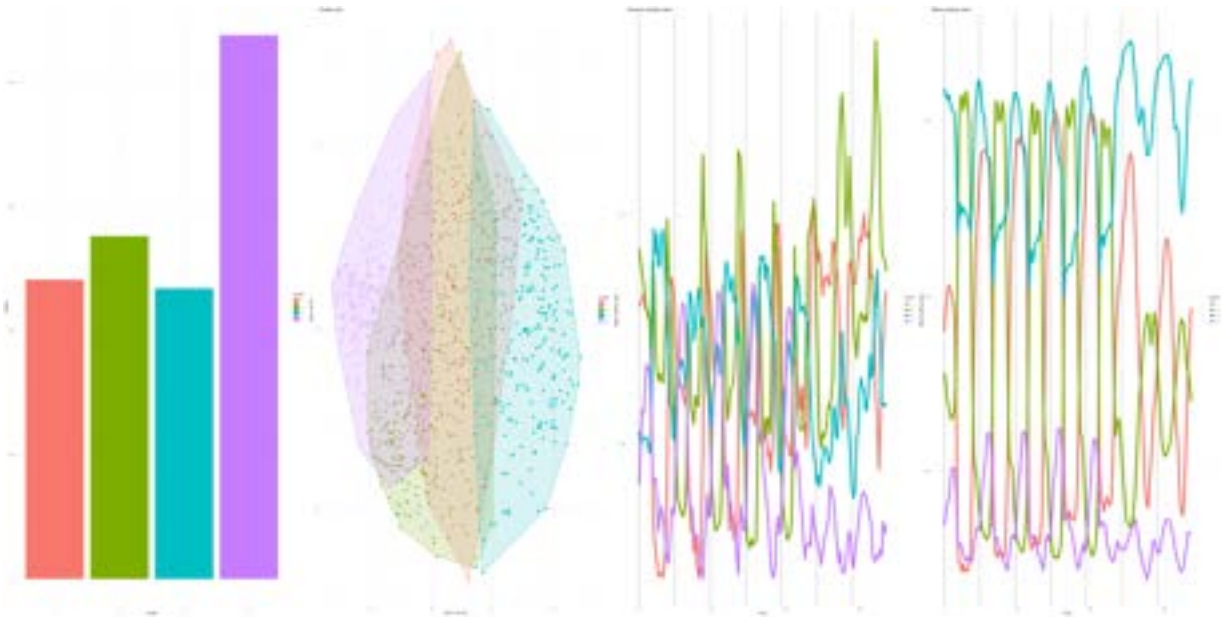
```
###############
#### k = 3 ####
###############

df = data.frame(cluster = c("1", "2", "3"), effectif = c(reskmeans3$size))

load1 = loading[reskmeans3$cluster==1,]
load2 = loading[reskmeans3$cluster==2,]
load3 = loading[reskmeans3$cluster==3,]

meanload1 = colMeans(load1)
meanload2 = colMeans(load2)
```

```
meanload3 = colMeans(load3)
dfmean = as.data.frame(meanload1)
dfmean$meanload2 = meanload2
dfmean$meanload3 = meanload3
time_range = 1:ncol(loading)
dfmean$time_range = time_range

varload1 = colVars(load1)
varload2 = colVars(load2)
varload3 = colVars(load3)
dfvar = as.data.frame(varload1)
dfvar$varload2 = varload2
dfvar$varload3 = varload3
time_range = 1:ncol(loading)
dfvar$time_range = time_range

p5 = ggplot(df, aes(x = cluster, y = effectif, fill=cluster)) +
    geom_bar(stat='identity') +
    theme_minimal()

p6 = fviz_cluster(reskmeans3, data=pca$ind$coord, ellipse.type="convex", lab

p7 = ggplot(dfvar,aes(x=time_range)) +
    geom_line(aes(y=varload1, color='1'), linewidth = 3) +
    geom_line(aes(y=varload2, color='2'), linewidth = 3) +
    geom_line(aes(y=varload3, color='3'), linewidth = 3) +
    labs(title = "Variance loading value", x = "Hour", y = "Mean loading val
    geom_vline(xintercept=time_tick, linetype="dashed") +
    theme_minimal()

p8 = ggplot(dfmean,aes(x=time_range)) +
    geom_line(aes(y=meanload1, color='1'), linewidth = 3) +
    geom_line(aes(y=meanload2, color='2'), linewidth = 3) +
    geom_line(aes(y=meanload3, color='3'), linewidth = 3) +
    labs(title = "Mean loading value", x = "Hour", y = "Mean loading value")
    geom_vline(xintercept=time_tick, linetype="dashed") +
    theme_minimal()

ggarrange(p5,p6,p7,p8, ncol = 4)
```



In [51]:
```
###############
#### k = 3 ####
###############

df = data.frame(cluster = c("1", "2", "3", "4"), effectif = c(reskmeans4$siz

load1 = loading[reskmeans4$cluster==1,]
load2 = loading[reskmeans4$cluster==2,]
load3 = loading[reskmeans4$cluster==3,]
load4 = loading[reskmeans4$cluster==4,]

meanload1 = colMeans(load1)
meanload2 = colMeans(load2)
meanload3 = colMeans(load3)
meanload4 = colMeans(load4)
dfmean = as.data.frame(meanload1)
dfmean$meanload2 = meanload2
dfmean$meanload3 = meanload3
dfmean$meanload4 = meanload4
time_range = 1:ncol(loading)
dfmean$time_range = time_range

varload1 = colVars(load1)
varload2 = colVars(load2)
varload3 = colVars(load3)
varload4 = colVars(load4)
dfvar = as.data.frame(varload1)
dfvar$varload2 = varload2
dfvar$varload3 = varload3
dfvar$varload4 = varload4
time_range = 1:ncol(loading)
dfvar$time_range = time_range

p9 = ggplot(df, aes(x = cluster, y = effectif, fill=cluster)) +
    geom_bar(stat='identity') +
    theme_minimal()
```

```
p10 = fviz_cluster(reskmeans4, data=pca$ind$coord, ellipse.type="convex", la

p11 = ggplot(dfvar,aes(x=time_range)) +
    geom_line(aes(y=varload1, color='1'), linewidth = 3) +
    geom_line(aes(y=varload2, color='2'), linewidth = 3) +
    geom_line(aes(y=varload3, color='3'), linewidth = 3) +
    geom_line(aes(y=varload4, color='4'), linewidth = 3) +
    labs(title = "Variance loading value", x = "Hour", y = "Mean loading val
    geom_vline(xintercept=time_tick, linetype="dashed") +
    theme_minimal()

p12 = ggplot(dfmean,aes(x=time_range)) +
    geom_line(aes(y=meanload1, color='1'), linewidth = 3) +
    geom_line(aes(y=meanload2, color='2'), linewidth = 3) +
    geom_line(aes(y=meanload3, color='3'), linewidth = 3) +
    geom_line(aes(y=meanload4, color='4'), linewidth = 3) +
    labs(title = "Mean loading value", x = "Hour", y = "Mean loading value")
    geom_vline(xintercept=time_tick, linetype="dashed") +
    theme_minimal()

ggarrange(p9,p10,p11,p12, ncol = 4)
```



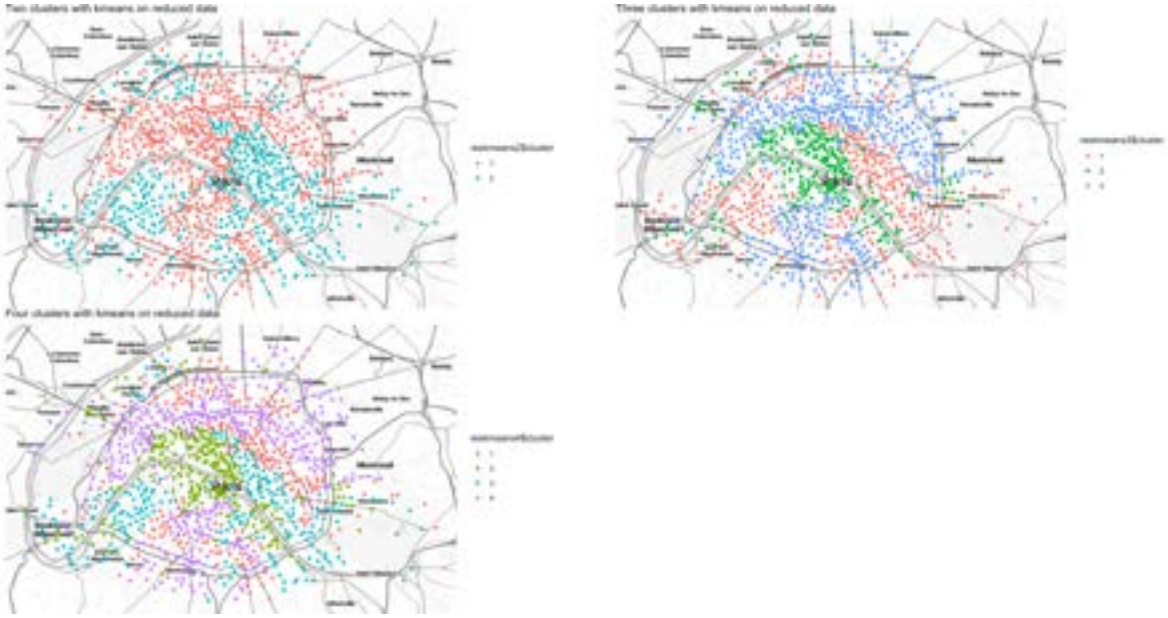
We can see that the clusters overlap a lot more when computed with the PCA data, making them less qualitative.

4.1.2.2. Visualization of clusters on the map

In [52]:
```
#hill = as.factor(velib$bonus)
options(repr.plot.width = 20, repr.plot.height = 10)

##############

reskmeans2$cluster = as.factor(reskmeans2$cluster)

df2 = data.frame(size=c(sum(reskmeans2$cluster==1), sum(reskmeans2$cluster==
                 labels = c('cluster 1','cluster 2'))
```

```
p2 = qmplot(data=coord, longitude, latitude, color=reskmeans2$cluster) +
    #scale_color_manual(values = c("1" = "cornflowerblue", "2" = "darkorange
    labs(title = 'Two clusters with kmeans on reduced data')

##############

reskmeans3$cluster = as.factor(reskmeans3$cluster)

df3 = data.frame(size=c(sum(reskmeans3$cluster==1), sum(reskmeans3$cluster==
                labels = c('cluster 1','cluster 2', 'cluster 3'))

p3 = qmplot(data=coord, longitude, latitude, color=reskmeans3$cluster) +
    #scale_color_manual(values = c("1" = "cornflowerblue", "2" = "darkorange
    labs(title = 'Three clusters with kmeans on reduced data')

##############

reskmeans4$cluster = as.factor(reskmeans4$cluster)

df4 = data.frame(size=c(sum(reskmeans4$cluster==1), sum(reskmeans4$cluster==
                labels = c('cluster 1','cluster 2', 'cluster 3', 'cluster 4'

p4 = qmplot(data=coord, longitude, latitude, color=reskmeans4$cluster) +
    #scale_color_manual(values = c("1" = "cornflowerblue", "2" = "darkorange
    labs(title = 'Four clusters with kmeans on reduced data')

ggpubr::ggarrange(p2,p3, p4)
```

```
ℹ Using `zoom = 12`

ℹ © Stadia Maps © Stamen Design © OpenMapTiles © OpenStreetMap contributors.

ℹ Using `zoom = 12`

ℹ © Stadia Maps © Stamen Design © OpenMapTiles © OpenStreetMap contributors.

ℹ Using `zoom = 12`

ℹ © Stadia Maps © Stamen Design © OpenMapTiles © OpenStreetMap contributors.
```

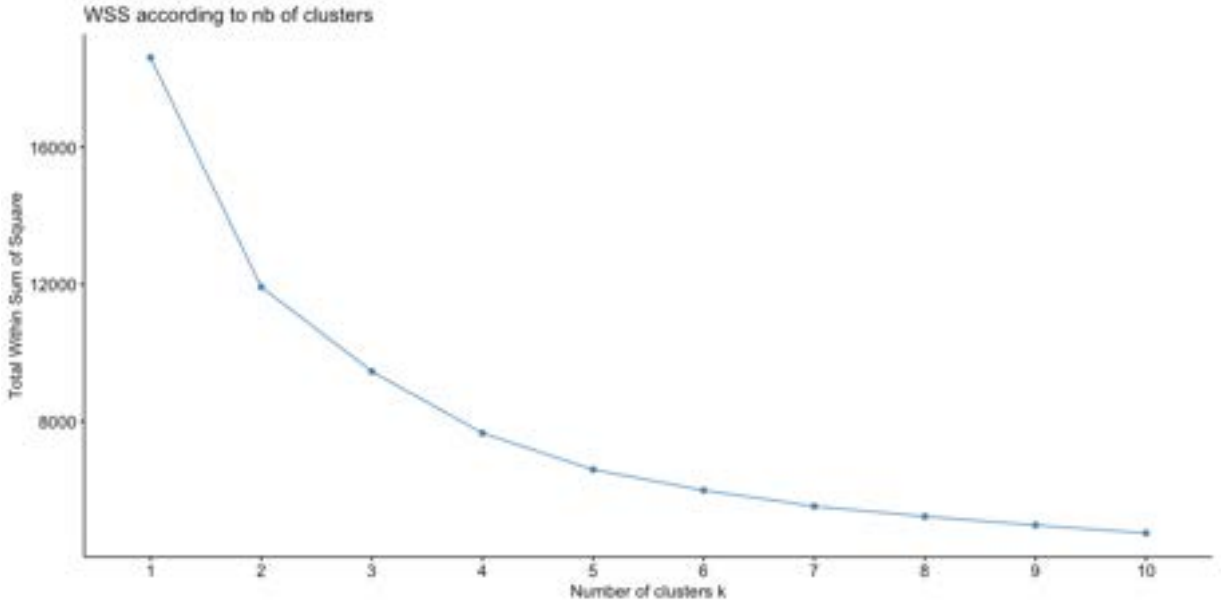


4.2. HCA clustering

4.2.1. Selection of the number of clusters

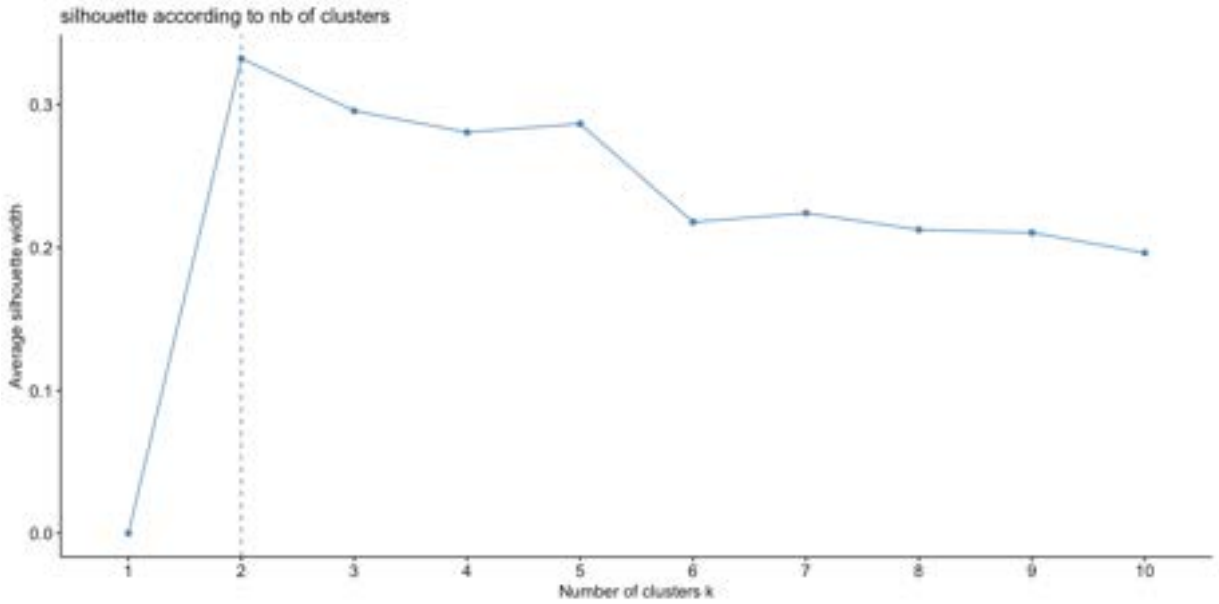
4.2.1.1. Determining the number of clusters using the total within sum of square metric

In [53]:
```
options(repr.plot.width = 12, repr.plot.height = 6)

fviz_nbclust(pca$ind$coord, FUNcluster=hcut, method="wss") + ggtitle("WSS ac
```



4.2.1.2. Determining the number of clusters using the silhouette metric

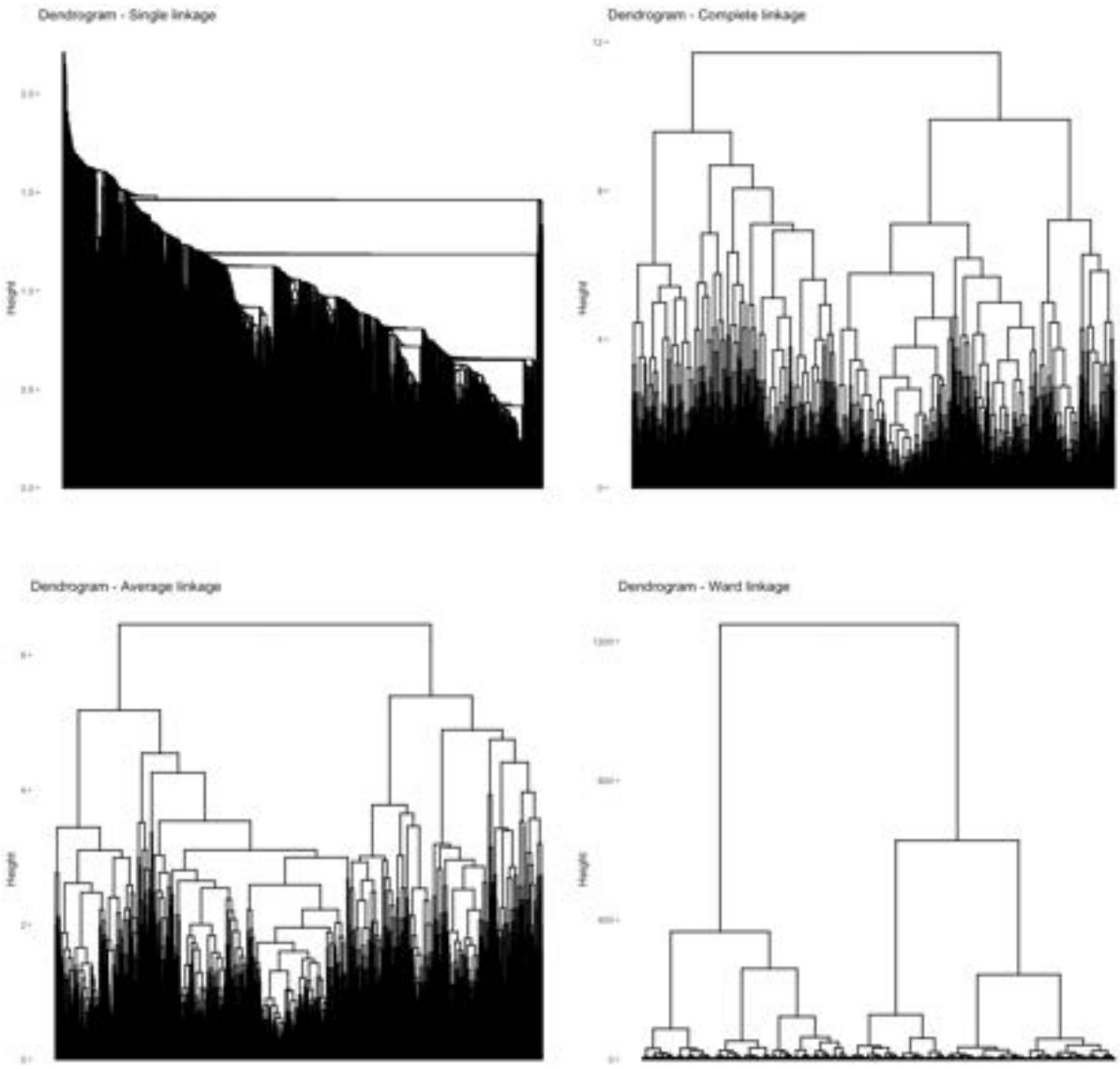
In [54]:
```
fviz_nbclust(pca$ind$coord, FUNcluster=hcut, method="silhouette") + ggtitle(
```



4.2.2. Visualization of different dendrograms and evaluation of the effect of the choice of the linkage function

4.2.2.1. Dendrogram with different linkage methods

In [55]:
```
d_pca = dist(pca$ind$coord, method="euclidean")
options(repr.plot.width=15, repr.plot.height=15)

hclustsingle = hclust(d_pca, method="single")
hclustcomplete = hclust(d_pca, method="complete")
hclustaverage = hclust(d_pca, method="average")
hclustward = hclust(d_pca, method="ward.D")

p1 <- fviz_dend(hclustsingle, show_labels=FALSE, main='Dendrogram - Single l
p2 <- fviz_dend(hclustcomplete, show_labels=FALSE, main='Dendrogram - Comple
p3 <- fviz_dend(hclustaverage, show_labels=FALSE, main='Dendrogram - Average
p4 <- fviz_dend(hclustward, show_labels=FALSE, main='Dendrogram - Ward linka

gridExtra::grid.arrange(p1, p2, p3, p4)
```



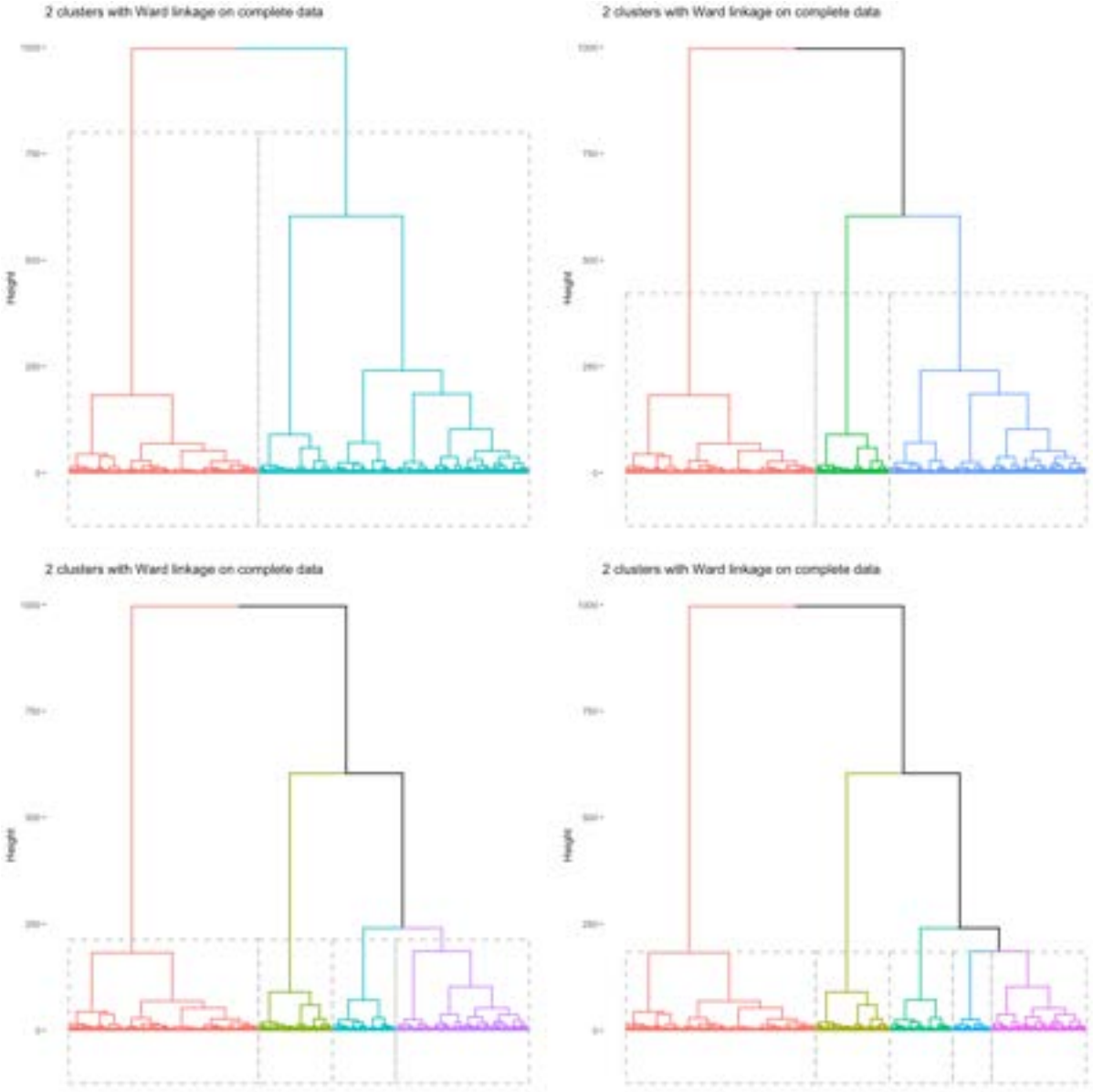
For some unknown reason we were unable to generate the dendrogram with single linkage for the complete data. It did however work with the reduced data (cf. Part 4)

4.2.2.2. Dendrogram with Ward linkage

In [56]:
```
d = dist(loading, method="euclidean")
options(repr.plot.width=15, repr.plot.height=15)

hclustward = hclust(d, method="ward.D")

p1 <- fviz_dend(hclustward, k=2, show_labels=FALSE, rect=TRUE, main='2 clust
print('p1 done :)')
p2 <- fviz_dend(hclustward, k=3, show_labels=FALSE, rect=TRUE, main='2 clust
print('p2 done :)')
p3 <- fviz_dend(hclustward, k=4, show_labels=FALSE, rect=TRUE, main='2 clust
print('p3 done :)')
p4 <- fviz_dend(hclustward, k=5, show_labels=FALSE, rect=TRUE, main='2 clust
print('p4 done :)')
```

```
#ggpubr::ggarrange(p1,p2,p3,p4)
gridExtra::grid.arrange(p1,p2,p3,p4)
```

```
[1] "p1 done :)"
[1] "p2 done :)"
[1] "p3 done :)"
[1] "p4 done :)"
```



4.2.3. Visualization and interpretation of k-means clusters

4.2.3.1. Four descriptive plots of cluster

In [57]:
```
hclustward2 = cutree(hclustward, k = 2)
hclustward3 = cutree(hclustward, k = 3)
hclustward4 = cutree(hclustward, k = 4)
hclustward5 = cutree(hclustward, k = 5)
```

In [58]:
```
time_tick = 1 + 24*(0:6)
```

```r
options(repr.plot.width = 50, repr.plot.height = 25)
#size=c(sum(hclustward2==1), sum(hclustward2==2))

###############
#### k = 2 ####
###############

df = data.frame(cluster = c("1", "2"), effectif = c(sum(hclustward2==1), sum

load1 = loading[hclustward2==1,]
load2 = loading[hclustward2==2,]

meanload1 = colMeans(load1)
meanload2 = colMeans(load2)
dfmean = as.data.frame(meanload1)
dfmean$meanload2 = meanload2
time_range = 1:ncol(loading)
dfmean$time_range = time_range

varload1 = colVars(load1)
varload2 = colVars(load2)
dfvar = as.data.frame(varload1)
dfvar$varload2 = varload2
time_range = 1:ncol(loading)
dfvar$time_range = time_range

p1 = ggplot(df, aes(x = cluster, y = effectif, fill=cluster)) +
     geom_bar(stat='identity') +
     labs(title = "Cluster frequency (k = 2)") +
     theme_minimal()

p2 = ggplot(pca$ind$coord, aes(x = coord$longitude, y = coord$latitude)) +
     geom_point(aes(color = factor(hclustward2)), size = 3) +
     labs(title = "HAC individuals scatter plot", x = "Dim 1 (40.5%)", y = "
     theme_minimal()

p3 = ggplot(dfvar,aes(x=time_range)) +
    geom_line(aes(y=varload1, color='1'), linewidth = 3) +
    geom_line(aes(y=varload2, color='2'), linewidth = 3) +
    labs(title = "Variance loading value", x = "Hour", y = "Mean loading val
    geom_vline(xintercept=time_tick, linetype="dashed") +
    theme_minimal()

p4 = ggplot(dfmean,aes(x=time_range)) +
    geom_line(aes(y=meanload1, color='1'), linewidth = 3) +
    geom_line(aes(y=meanload2, color='2'), linewidth = 3) +
    labs(title = "Mean loading value", x = "Hour", y = "Mean loading value")
    geom_vline(xintercept=time_tick, linetype="dashed") +
    theme_minimal()

ggarrange(p1,p2,p3,p4, ncol = 4)
```