DATA SCIENCE METHODS

Assignment 01

09/03/2022

Group 01:

Bilge Kasapoğlu – u941664

Hoan Van Nguyen – u1274449

Jiahe Wang – u489199

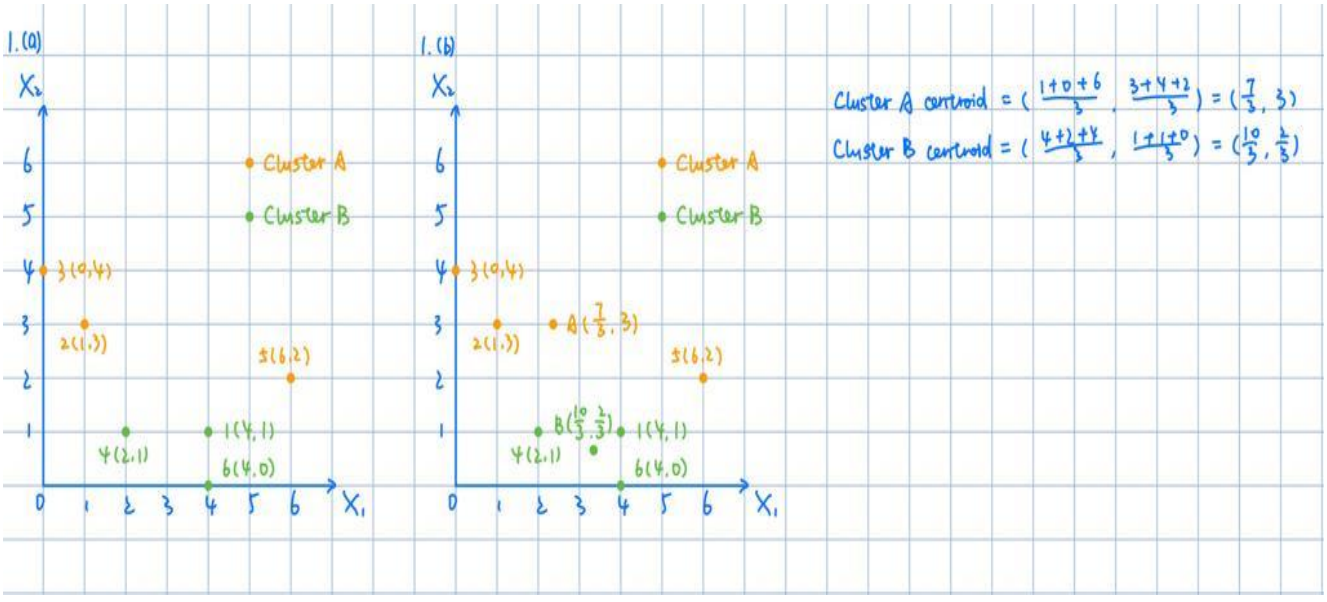
Roshini Sudhaharan – u725261

QUESTION 01

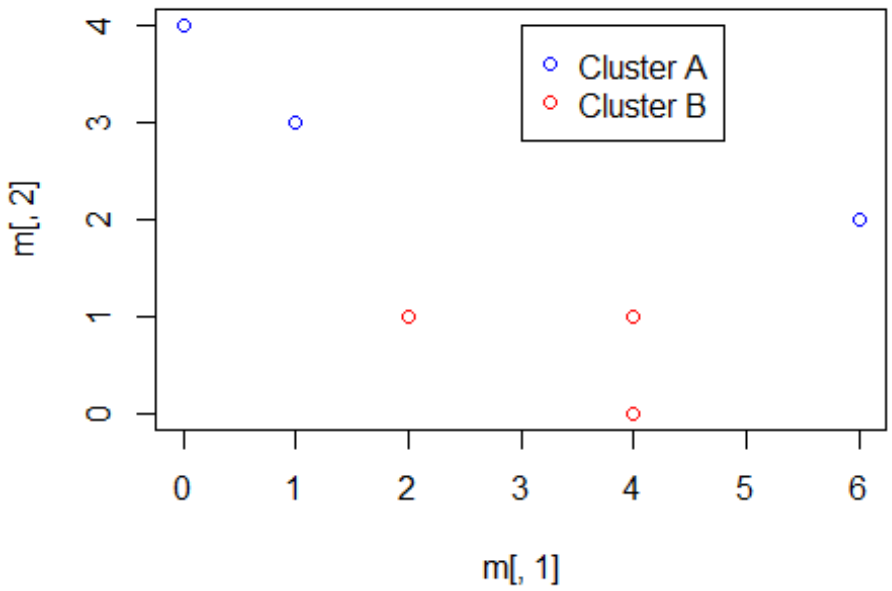
```
#create matrix with n = 6 observations and p = 2 features;
#third column indicating cluster: 1 is cluster A, 2 is cluster B
set.seed(1)
m = cbind(x1 = c(4, 1, 0, 2, 6, 4), x2 = c(1, 3, 4, 1, 2, 0), clusters =
c(2,1,1,2,1,2))
m
```

```
##      x1 x2 clusters
## [1,]  4  1        2
## [2,]  1  3        1
## [3,]  0  4        1
## [4,]  2  1        2
## [5,]  6  2        1
## [6,]  4  0        2
```

a.

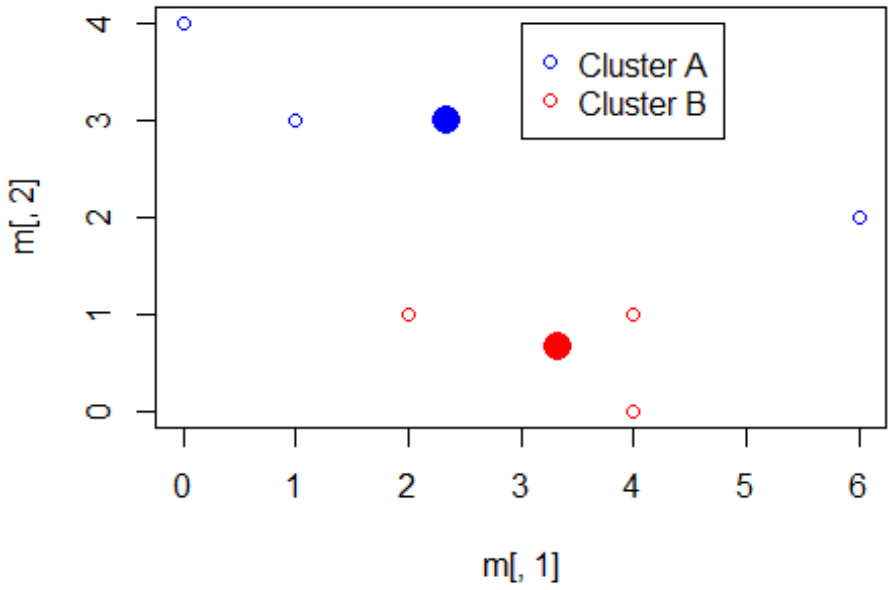


```r
#plot x1 against x2

plot(m[,1], m[,2], col = ifelse(m[,3] == 1, "blue", "red"))
legend(3,4,legend = c("Cluster A", "Cluster B"), col = c("Blue", "Red"), pch = 1)
```

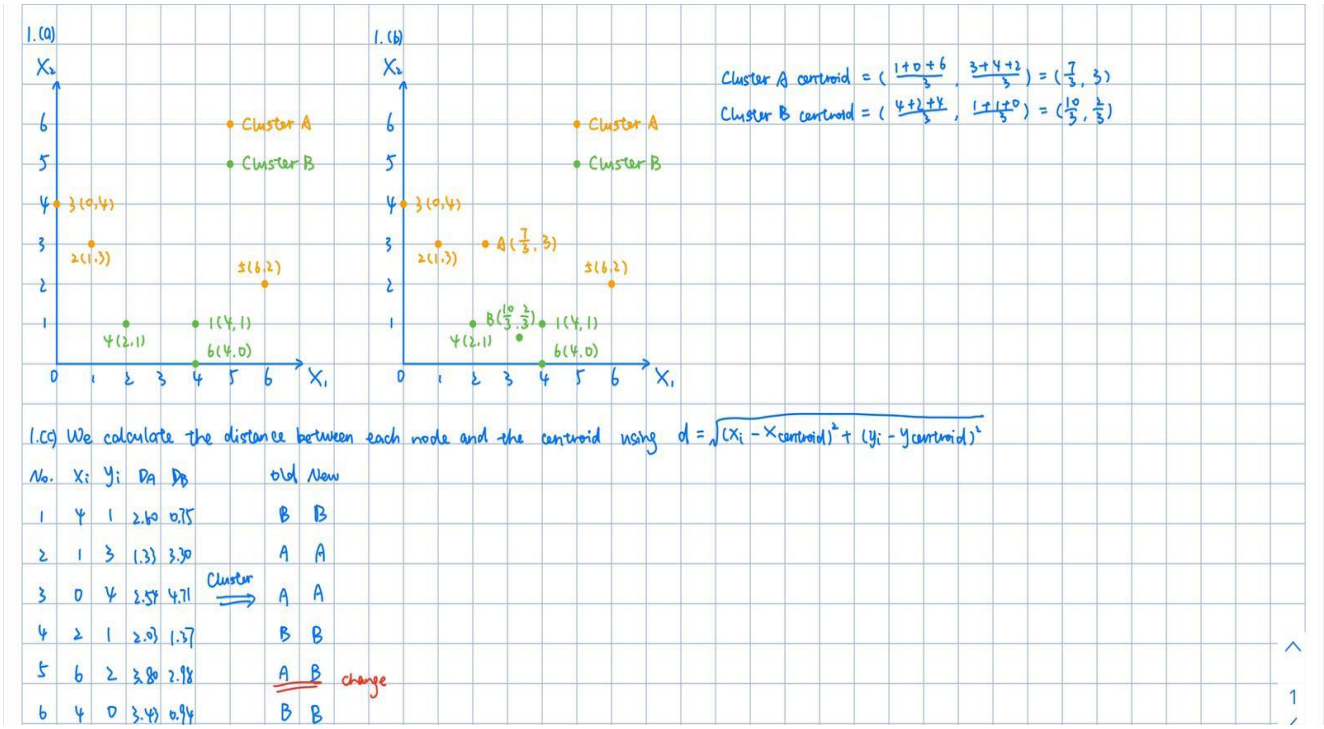


b.

```r
#calculate centroids for two clusters A & B and plot them

centroidA = c(mean(m[m[,3]==1, 1]), mean(m[m[,3]==1, 2]))
centroidB = c(mean(m[m[,3]==2, 1]), mean(m[m[,3]==2, 2]))
print(centroidA)

## [1] 2.333333 3.000000

print(centroidB)

## [1] 3.3333333 0.6666667

plot(m[,1], m[,2], col = ifelse(m[,3] == 1, "blue", "red"))
legend(3,4,legend = c("Cluster A", "Cluster B"), col = c("Blue", "Red"), pch = 1)
points(x = centroidA[1], y = centroidA[2], col = "blue", pch = 16, cex = 2)
points(x = centroidB[1], y = centroidB[2], col = "red", pch = 16, cex = 2)
```



c.




```
#assign each observations to the centroid
#to which it is the closest in terms of Euclidean distance

euclidean = function(a, b) {
  return(sqrt((a[1] - b[1])^2 + (a[2]-b[2])^2))
}
assign_clusters = function(m, centroidA, centroidB) {
  new_clusters = rep(NA, nrow(m))
  for (i in 1:nrow(m)) {
```

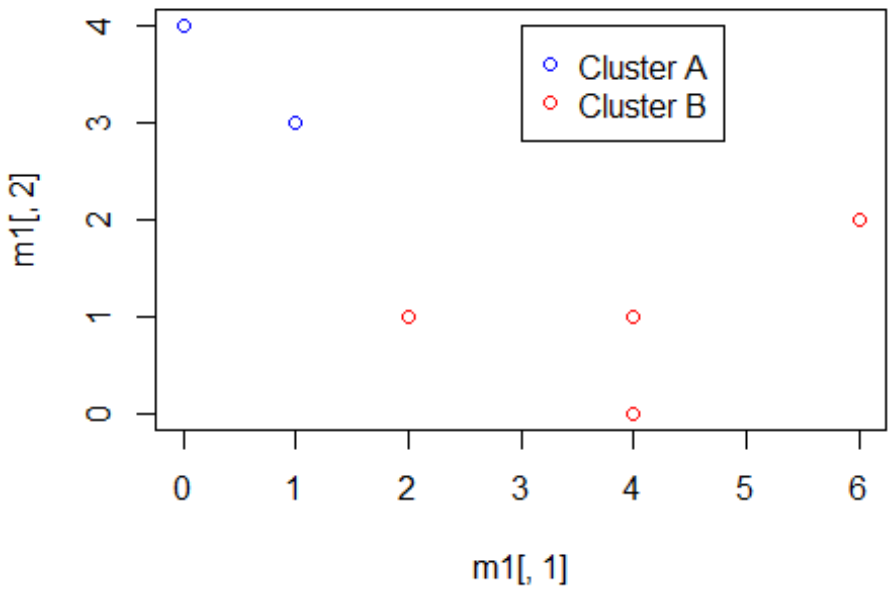
```
    if (euclidean(m[i,], centroidA) < euclidean(m[i,], centroidB)) {
      new_clusters[i] = 1
    } else {
      new_clusters[i] = 2
    }
  }
  return(new_clusters)
}
new_clusters = assign_clusters(m, centroidA, centroidB)
new_clusters
```

```
## [1] 2 1 1 2 2 2
```

```
#new matrix including new cluster membership

m1 <- cbind(m, new_clusters)
m1
```

```
##      x1 x2 clusters new_clusters
## [1,]  4  1        2            2
## [2,]  1  3        1            1
## [3,]  0  4        1            1
## [4,]  2  1        2            2
## [5,]  6  2        1            2
## [6,]  4  0        2            2
```

```
#new plot showing new cluster membership

plot(m1[,1], m1[,2], col = ifelse(m1[,4] == 1, "blue", "red"))
legend(3,4,legend = c("Cluster A", "Cluster B"), col = c("Blue", "Red"), pch = 1)
```



d.

```
#repeat 1c until the answer stops changing

last_clusters = rep(-1, 6)
while (!all(last_clusters == new_clusters)) {
  last_clusters = new_clusters
  centroidA = c(mean(m1[m1[,4] == 1, 1]), mean(m1[m1[,4] == 1, 2]))
  centroidB = c(mean(m1[m1[,4] == 2, 1]), mean(m1[m1[,4] == 2, 2]))
  print(centroidA)
  print(centroidB)
  new_clusters = assign_clusters(m1, centroidA, centroidB)
}

## [1] 0.5 3.5
## [1] 4 1

new_clusters

## [1] 2 1 1 2 2 2
```

According to the results above, we can conclude that observations 2 and 3 belong to one cluster, and other observations 1, 4, 5, and 6 belong to another cluster.

QUESTION 02

```
# Create a directory to store downloaded data
dir.create("A1/data")

## Warning in dir.create("A1/data"): cannot create dir 'A1\data', reason 'No such
## file or directory'

# Download data
download_data <- function(url, filename){
  download.file(url = url, destfile = paste0(filename, ".csv"))
}

url <- "https://drive.google.com/uc?id=1DaYBBo_qohz-
QiMOIPu08m0QXbjRsu0Q&export=download"

download_data(url, "2020_NL_Region_Mobility_Report")

# Load the data
dta <- read.csv("2020_NL_Region_Mobility_Report.csv")

# cleaning the dataset
d = dta[!(is.na(dta$sub_region_1) | dta$sub_region_1 =="") &
(is.na(dta$sub_region_2) | dta$sub_region_2==""),]
d <- d %>% select(date,sub_region_1,sub_region_2,
                  transit_stations_percent_change_from_baseline,
                  workplaces_percent_change_from_baseline,
)
```

```r
# reshaping the dataset

transit = d %>% pivot_wider(
  id_cols = "date",
  names_from = "sub_region_1",names_prefix = "transit",
  names_sep = "_",
  values_from = c(transit_stations_percent_change_from_baseline),
)

work = d %>% pivot_wider(
  id_cols = "date",
  names_from = "sub_region_1",names_prefix = "work",
  names_sep = "_",
  values_from = c(workplaces_percent_change_from_baseline),
)



combined = merge(transit,work,by='date',all=T)
combined$date=as.Date(combined$date)
```

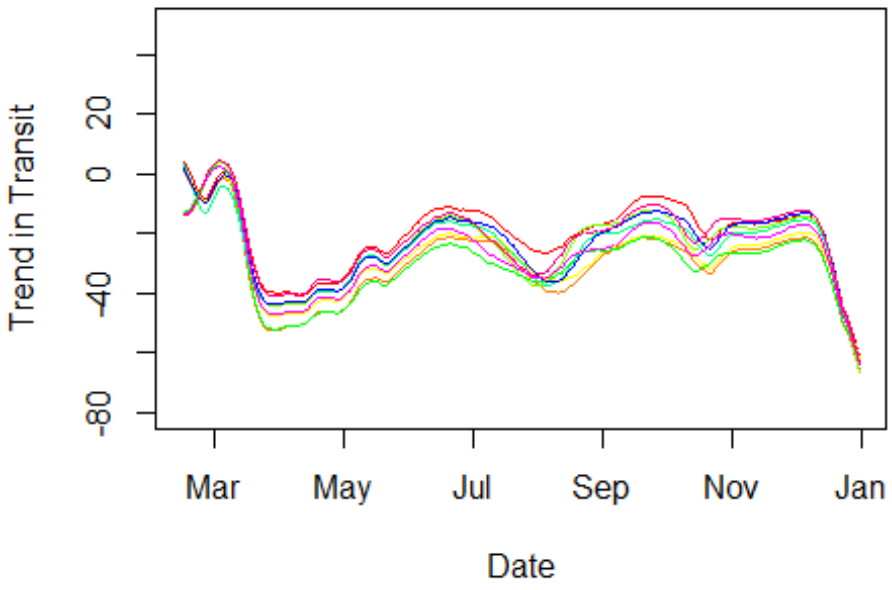
a.

```r
combined_new = combined[,1] %>% as.data.frame()
names(combined_new)[1] <- "date"
regions <- names(combined)[2:ncol(combined)]
date = combined[,1]  %>% as.Date
for (i in c(2:ncol(combined))){
  ts1 <- combined[,c(1,i)]
  ts1 = na.omit(ts1)
  ts = ts1[,2]
  smoothed <- hpfilter(ts, freq=200)
  ts1[,2] = as.matrix(smoothed$trend)
  combined_new = combined_new
  combined_new <- merge(ts1, combined_new, by = "date", all.y = TRUE)
  names(combined_new)[ncol(combined_new)] <- regions[i-1]
}
summary(combined_new)
```

```
##       date              workZeeland.V1      workUtrecht.V1
##  Min.   :2020-02-15   Min.   :-61.15221   Min.   :-66.16620
##  1st Qu.:2020-05-05   1st Qu.:-25.78338   1st Qu.:-37.30416
##  Median :2020-07-24   Median :-17.13599   Median :-27.35050
##  Mean   :2020-07-24   Mean   :-20.24214   Mean   :-30.03427
##  3rd Qu.:2020-10-12   3rd Qu.:-12.50645   3rd Qu.:-22.49182
##  Max.   :2020-12-31   Max.   :  4.22166   Max.   :  0.73892
##                       NA's   :3
##  workSouth Holland.V1  workOverijssel.V1  workNorth Holland.V1
##  Min.   :-61.82493    Min.   :-66.26223   Min.   :-63.37622
##  1st Qu.:-34.14222    1st Qu.:-32.43427   1st Qu.:-36.13931
##  Median :-25.56455    Median :-20.24331   Median :-28.14503
##  Mean   :-27.83801    Mean   :-23.83178   Mean   :-30.21807
##  3rd Qu.:-21.33089    3rd Qu.:-15.73630   3rd Qu.:-24.27495
##  Max.   :  1.44974    Max.   :  3.71588   Max.   :  2.69142
```

```
##
##  workNorth Brabant.V1   workLimburg.V1     workGroningen.V1
##  Min.   :-63.30115    Min.   :-60.83267   Min.   :-63.32686
##  1st Qu.:-32.52579    1st Qu.:-30.05247   1st Qu.:-33.94933
##  Median :-21.30181    Median :-19.02152   Median :-23.49639
##  Mean   :-24.91900    Mean   :-23.13396   Mean   :-25.93769
##  3rd Qu.:-17.19576    3rd Qu.:-15.48453   3rd Qu.:-18.04952
##  Max.   :  3.12409    Max.   :  4.16776   Max.   :  4.12526
##
##   workGelderland.V1   workFriesland.V1    workFlevoland.V1
##  Min.   :-63.45007   Min.   :-63.28986   Min.   :-63.69390
##  1st Qu.:-32.01014   1st Qu.:-30.94973   1st Qu.:-32.95734
##  Median :-19.78411   Median :-18.86011   Median :-25.18019
##  Mean   :-23.42368   Mean   :-22.19315   Mean   :-26.61321
##  3rd Qu.:-15.36001   3rd Qu.:-14.64459   3rd Qu.:-19.59444
##  Max.   :  2.11215   Max.   :  4.69689   Max.   :  2.41037
##                                          NA's   :3
##    workDrenthe.V1     transitZeeland.V1   transitUtrecht.V1
##  Min.   :-63.38117   Min.   :-58.75816   Min.   :-74.17694
##  1st Qu.:-30.09956   1st Qu.:-36.89267   1st Qu.:-58.27816
##  Median :-19.55987   Median :-26.33716   Median :-51.93085
##  Mean   :-22.00000   Mean   :-20.82432   Mean   :-49.43925
##  3rd Qu.:-14.18033   3rd Qu.:  1.87876   3rd Qu.:-45.05694
##  Max.   :  4.37224   Max.   : 16.74319   Max.   :  7.53359
##  NA's   :3           NA's   :25
##  transitSouth Holland.V1 transitOverijssel.V1 transitNorth Holland.V1
##  Min.   :-64.19970       Min.   :-67.51329    Min.   :-74.33453
##  1st Qu.:-47.22250       1st Qu.:-48.88477    1st Qu.:-60.76767
##  Median :-43.46913       Median :-41.15154    Median :-55.08408
##  Mean   :-41.07165       Mean   :-39.66355    Mean   :-50.57944
##  3rd Qu.:-36.31919       3rd Qu.:-34.46636    3rd Qu.:-42.78297
##  Max.   :  0.95582       Max.   :  3.86257    Max.   :  1.62485
##
##  transitNorth Brabant.V1  transitLimburg.V1  transitGroningen.V1
##  Min.   :-68.40679       Min.   :-59.93348   Min.   :-64.37397
##  1st Qu.:-51.64226       1st Qu.:-38.04448   1st Qu.:-47.01244
##  Median :-46.84983       Median :-31.98389   Median :-39.43008
##  Mean   :-44.16199       Mean   :-30.48598   Mean   :-38.32087
##  3rd Qu.:-39.77811       3rd Qu.:-21.72211   3rd Qu.:-34.01597
##  Max.   : 10.16420       Max.   :  7.70066   Max.   :  5.27273
##
##  transitGelderland.V1 transitFriesland.V1 transitFlevoland.V1
##  Min.   :-61.27717    Min.   :-52.77479   Min.   :-66.14571
##  1st Qu.:-42.86606    1st Qu.:-28.73414   1st Qu.:-44.77956
##  Median :-37.18129    Median :-14.04728   Median :-37.02641
##  Mean   :-35.62617    Mean   :-13.90159   Mean   :-37.27987
##  3rd Qu.:-29.68818    3rd Qu.:  2.86600   3rd Qu.:-30.94921
##  Max.   :  1.91965    Max.   : 20.24478   Max.   :  3.54422
##                       NA's   :6           NA's   :3
##    workZeeland.V1
##  Min.   :-61.37160
##  1st Qu.:-38.08222
```

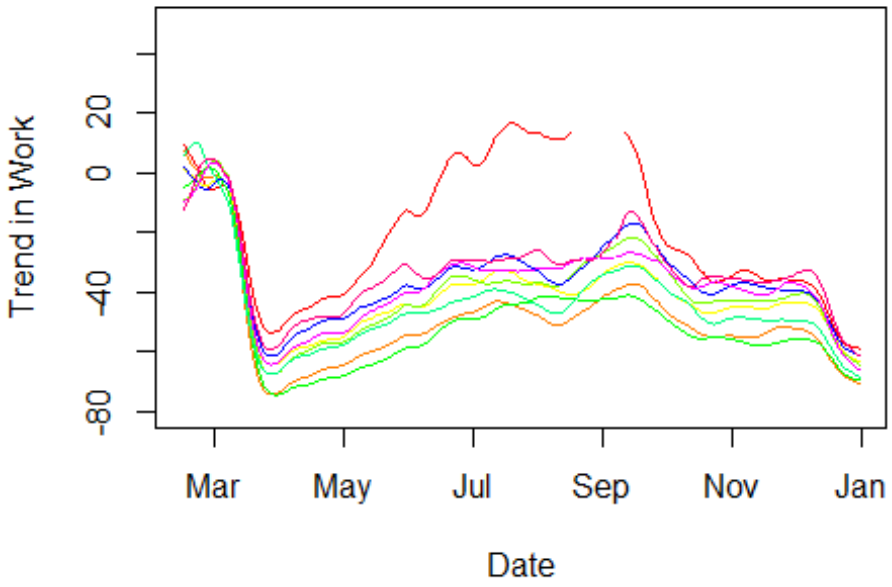
```
##  Median :-33.98673
##  Mean   :-33.06984
##  3rd Qu.:-28.93562
##  Max.   :  4.71238
##  NA's   :6

names(combined_new)[2:ncol(combined_new)] = regions

transit_hp = combined_new[,c(1:ncol(transit))]
work_hp = select(combined_new,c(date, workDrenthe:workZeeland))

cl <- rainbow(12)

plot(transit_hp$date, transit_hp$transitDrenthe, type="l", col = cl[1], ylim = c(-
80,50), xlab = "Date", ylab = "Trend in Transit")
lines(transit_hp$date, transit_hp$transitFlevoland, type = "l", col = cl[2])
lines(transit_hp$date, transit_hp$transitFriesland, type = "l", col = cl[3])
lines(transit_hp$date, transit_hp$transitGelderland, type = "l", col = cl[4])
lines(transit_hp$date, transit_hp$transitGroningen, type = "l", col = cl[5])
lines(transit_hp$date, transit_hp$transitLimburg, type = "l", col = cl[6])
lines(transit_hp$date, transit_hp$transitNorthBrabant, type = "l", col = cl[7])
lines(transit_hp$date, transit_hp$transitNorthHolland, type = "l", col = cl[8])
lines(transit_hp$date, transit_hp$transitOverijssel, type = "l", col = cl[9])
lines(transit_hp$date, transit_hp$transitSouthHolland, type = "l", col = cl[10])
lines(transit_hp$date, transit_hp$transitUtrecht, type = "l", col = cl[11])
lines(transit_hp$date, transit_hp$transitZeeland, type = "l", col = cl[12])
```



We see a comovement in transit among different regions.

```
plot(work_hp$date, work_hp$workDrenthe, type="l", col = cl[1], ylim = c(-80,50),
xlab = "Date", ylab = "Trend in Work")
lines(work_hp$date, work_hp$workFlevoland, type = "l", col = cl[2])
lines(work_hp$date, work_hp$workFriesland, type = "l", col = cl[3])
```

```
lines(work_hp$date, work_hp$workGelderland, type = "l", col = cl[4])
lines(work_hp$date, work_hp$workGroningen, type = "l", col = cl[5])
lines(work_hp$date, work_hp$workLimburg, type = "l", col = cl[6])
lines(work_hp$date, work_hp$workNorthBrabant, type = "l", col = cl[7])
lines(work_hp$date, work_hp$workNorthHolland, type = "l", col = cl[8])
lines(work_hp$date, work_hp$workOverijssel, type = "l", col = cl[9])
lines(work_hp$date, work_hp$workSouthHolland, type = "l", col = cl[10])
lines(work_hp$date, work_hp$workUtrecht, type = "l", col = cl[11])
lines(work_hp$date, work_hp$workZeeland, type = "l", col = cl[12])
```



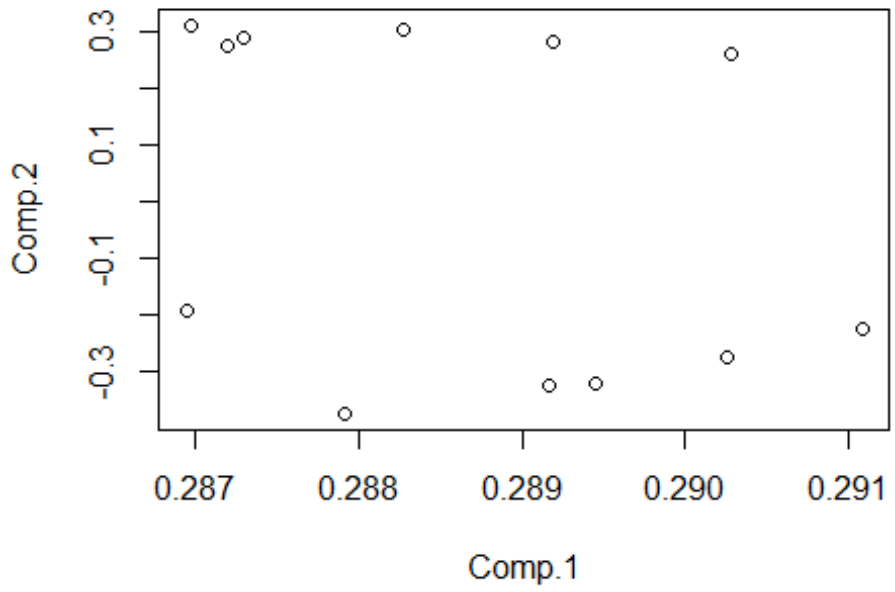
We see a comovement in work among different regions. However, we think the comovement is weaker in comparison to transit.

b.

```
t <- transit_hp[,2:ncol(transit_hp)]
t <- na.omit(t)
t <- scale(t)
pc_transit<-princomp(t,cor=TRUE,scores=TRUE)

loadings_transit_2 <- pc_transit$loadings[,1:2]
loadings_transit_2
```

```
##                          Comp.1     Comp.2
## transitDrenthe       0.2872077  0.2736024
## transitFlevoland     0.2882801  0.3007488
## transitFriesland     0.2869774  0.3093989
## transitGelderland    0.2902589 -0.2750322
## transitGroningen     0.2869498 -0.1930421
## transitLimburg       0.2891917  0.2803822
## transitNorth Brabant 0.2873059  0.2891416
## transitNorth Holland 0.2891703 -0.3223546
## transitOverijssel    0.2902784  0.2606661
## transitSouth Holland 0.2879154 -0.3724490
```

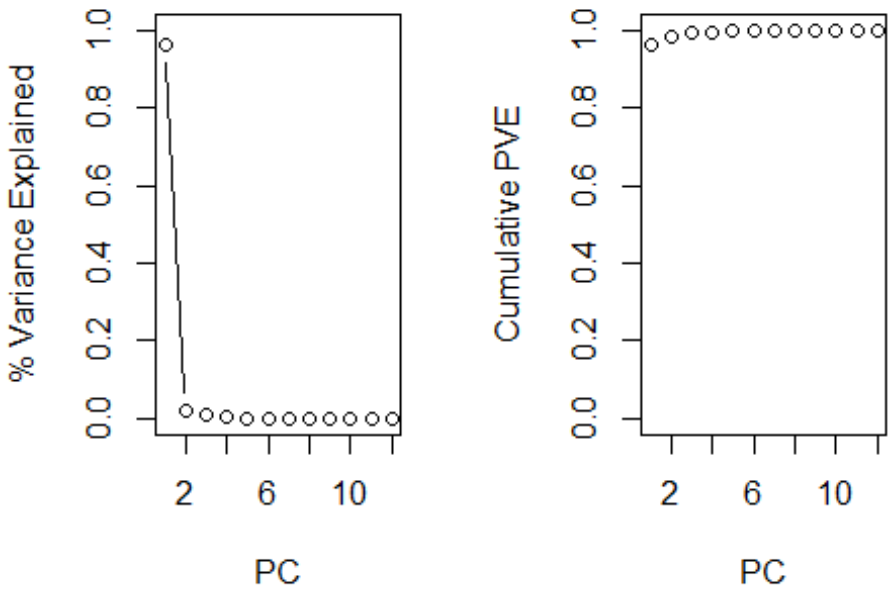
```
## transitUtrecht       0.2910786 -0.2249646
## transitZeeland       0.2894484 -0.3203945

PC1 <- loadings_transit_2[2]
PC2 <- loadings_transit_2[3]

checkPC1 <- as.data.frame(loadings_transit_2) %>%
arrange(abs(loadings_transit_2[,1]))
checkPC2 <- as.data.frame(loadings_transit_2) %>%
arrange(abs(loadings_transit_2[,2]))

# loadings plot
plot(loadings_transit_2)
```



We see that provinces cluster together. It seems like they cluster together on the second loading.

c.

```
pr.var=pc_transit$sdev^2
pve = pr.var/sum(pr.var)

# Put two plots side by side
par(mfrow=c(1,2))
plot(pve,xlab="PC",ylab="% Variance Explained",ylim=c(0,1),type='b')
plot(cumsum(pve),xlab="PC",ylab="Cumulative PVE",ylim=c(0,1),type='b')
```



The first two components seem to explain the most of the variation in the data.Thus, the first two components seems to be the appropriate solution for the PCA problem.

d.

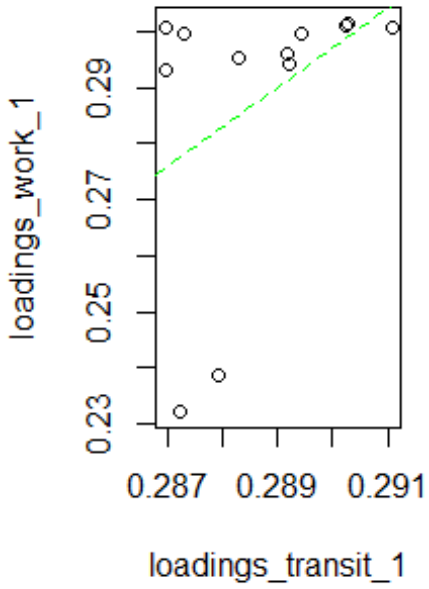
```
# The first component of the transit
loadings_transit_1 <- pc_transit$loadings[,1]

# The first component of the work
w <- work_hp[,2:ncol(work_hp)]
w <- na.omit(w)
w <- scale(w)
pc_work<-princomp(w,cor=TRUE,scores=TRUE)

loadings_work_1 <- pc_work$loadings[,1]
loadings_work_1
```

```
##       workDrenthe     workFlevoland     workFriesland    workGelderland
##         0.2321948         0.2952406         0.3006682         0.3010604
##     workGroningen       workLimburg workNorth Brabant workNorth Holland
##         0.2929839         0.2940664         0.2997617         0.2959767
##    workOverijssel workSouth Holland       workUtrecht       workZeeland
##         0.3014886         0.2385959         0.3007852         0.2996846
```

```
# The first component of the whole
whole <- combined_new[,2:ncol(combined_new)]
whole <- na.omit(whole)
whole <- scale(whole)

pc_whole<-princomp(whole,cor=TRUE,scores=TRUE)

loadings_whole_1 <- pc_whole$loadings[,1]
loadings_whole_1
```

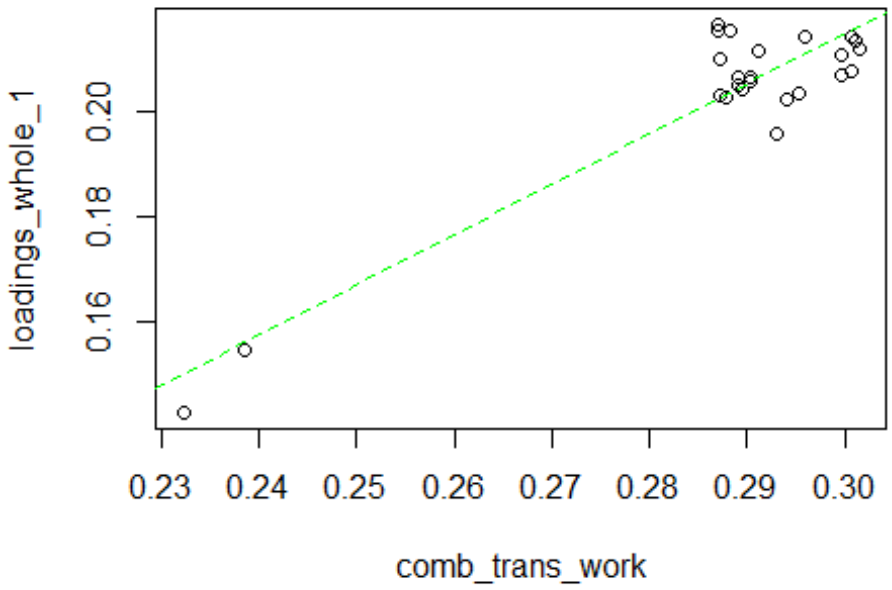
```
##       transitDrenthe     transitFlevoland     transitFriesland
##            0.2101852            0.2152953            0.2152513
##    transitGelderland     transitGroningen       transitLimburg
##            0.2057773            0.2165807            0.2051754
## transitNorth Brabant transitNorth Holland    transitOverijssel
##            0.2029695            0.2064625            0.2064379
## transitSouth Holland       transitUtrecht       transitZeeland
##            0.2026282            0.2117402            0.2041070
##          workDrenthe        workFlevoland        workFriesland
##            0.1429487            0.2034520            0.2076602
##       workGelderland        workGroningen          workLimburg
##            0.2133238            0.1958040            0.2023335
##    workNorth Brabant    workNorth Holland       workOverijssel
##            0.2071205            0.2143730            0.2120688
##    workSouth Holland          workUtrecht          workZeeland
##            0.1549028            0.2141133            0.2107004
```

```r
plot(loadings_transit_1, loadings_work_1, type ="p")

comb_trans_work <- append(loadings_transit_1, loadings_work_1)

par(mfrow = c(1, 2))
# the plot : work vs transit
plot(loadings_transit_1, loadings_work_1, type ="p")
abline(lm(loadings_work_1 ~ loadings_transit_1), lty = 2, col = "green")
```



It seems like the first PCs of Transit and Work do not seem to comove with each other.

```r
# the plot : work + transit vs whole
plot(comb_trans_work, loadings_whole_1, type ="p")
abline(lm(loadings_whole_1 ~ comb_trans_work), lty = 2, col = "green")
```



This is the plot of the first PCs of Work and Transit vs the combined data. It seems like they comove with each other pretty much. It makes more intuitive sense that the second one has comovement because the points are close to the green line (45 degrees line). We do not observe the same thing from the first plot.

e.

```
# Explained variance in %  - transit
pr.var=pc_transit$sdev^2
pve = pr.var/sum(pr.var) # the first PC explains 96% var. in the data


# Explained variance in % - work
pr.var_work=pc_work$sdev^2
pve_work = pr.var_work/sum(pr.var_work) # the first PC explains 89% var. in the
data

# Explained variance in % - combined
pr.var_combined=pc_whole$sdev^2
pve_combined = pr.var_combined/sum(pr.var_combined) # the first PC explains 85%
var. in the data
```

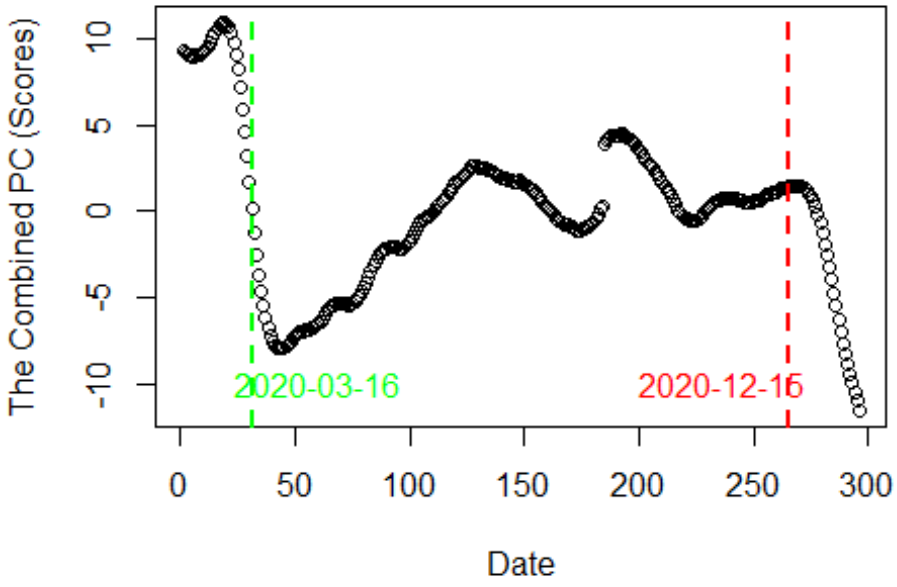
We think that work and transit are closer to a consistent estimate of the true underlying mobility since their first PC explains a higher variation in the data in comparison to the combined one.

When we compare work and transit, transit seems to be closer to the consistent estimate since it explains a higher variation in the data.

f.

```
# The scores of the first component of the whole
scores_whole_1 <- pc_whole$scores[,1]
```

```r
# the plot : the combined PC from the whole dataset over time, with two lockdowns
plot(1:296, scores_whole_1, type ="p", xlab = "Date", ylab = "The Combined PC
(Scores)")
abline(v = c(31, 265), lty = c(2, 2), lwd = c(2, 2), col = c("green", "red"))
text(c(60, 236), c(-10, -10), c("2020-03-16", "2020-12-15"), col = c("green",
"red"))
```



The first lockdown on 2020-03-16 was most successful. In the second one, it was already low compared to the first one, so the marginal effect was much stronger in the first one.

QUESTION 03

Assumptions.

① $p < n$: No. of variables is large but still smaller than sample size.

② rank $(x) = p$: If rank $(x) < p$ it means some variables are linear combo of others.

Singular value decomposition of X.

- X can be written as, $X = U.D.A' = ZA'$

The principle component (PCs) are linear combo of Xs, the Xs are also linear combo of the PCs.

Thus, $[\underbrace{Z_1, Z_2 \dots Z_p}_{Z}] = X [\underbrace{a_1 \dots a_p}_{A \rightarrow \text{matrix of loadings}}]$

$\Rightarrow Z = XA$

So,

$X = Z.A'$

Now, consider covariates X_j for $j = 1 \dots, p$

$X_j = Z_i a_{1j} + \underbrace{Z_2 a_{2j} + \dots + Z_p a_{pj}}_{\text{error}}$

Note that these PCs & loadings are orthogonal to each other.

Intuition: The algorithm relies on the fact that the PCs are orthogonal to each other which means we can separately solve the problem for the first PC and then subtract the linear combo of Zs from X to get some error. This error by definition will be orthogonal to second PC.

Steps:

① Standardize X

② Initialise the first PC (Z_1^a)

$$Z_1^a = X_1$$

③ Regress each column of X, X_j on Z_1^a to get estimate of the loadings:

$$a_{j1}^a = \frac{Z_1^{a'} X_j}{Z_1^{a'} Z_1^a}$$

④ Standardize loadings:

$$a_{j1}^b = \frac{a_{j1}^a}{\sqrt{a_1^{a'} a_1^a}}$$

To improve score estimates, regress every row of X, rx_i on a_1^b:

$$Z_{i1}^b = \frac{rx_i' a_1^b}{a_1^{b'} a_1^b} = rx_i' a_1^b$$

Substitute above Z_1^b instead of Z_1^a & iterate until the difference in Z_1's between 2 steps is very small.

To calculate Z_2, let

$$E = X - Z_1 a_1'$$

Repeat above steps for Z_2 instead of Z_1 & E instead of X. Iterate until all PCs are calculated.