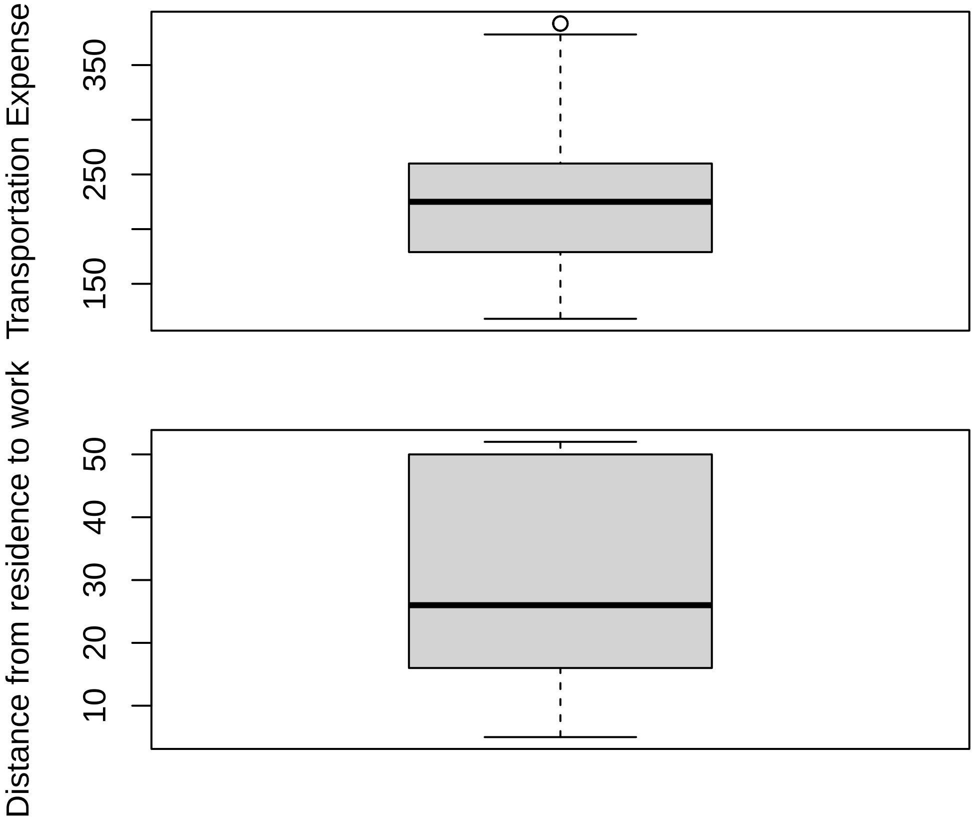
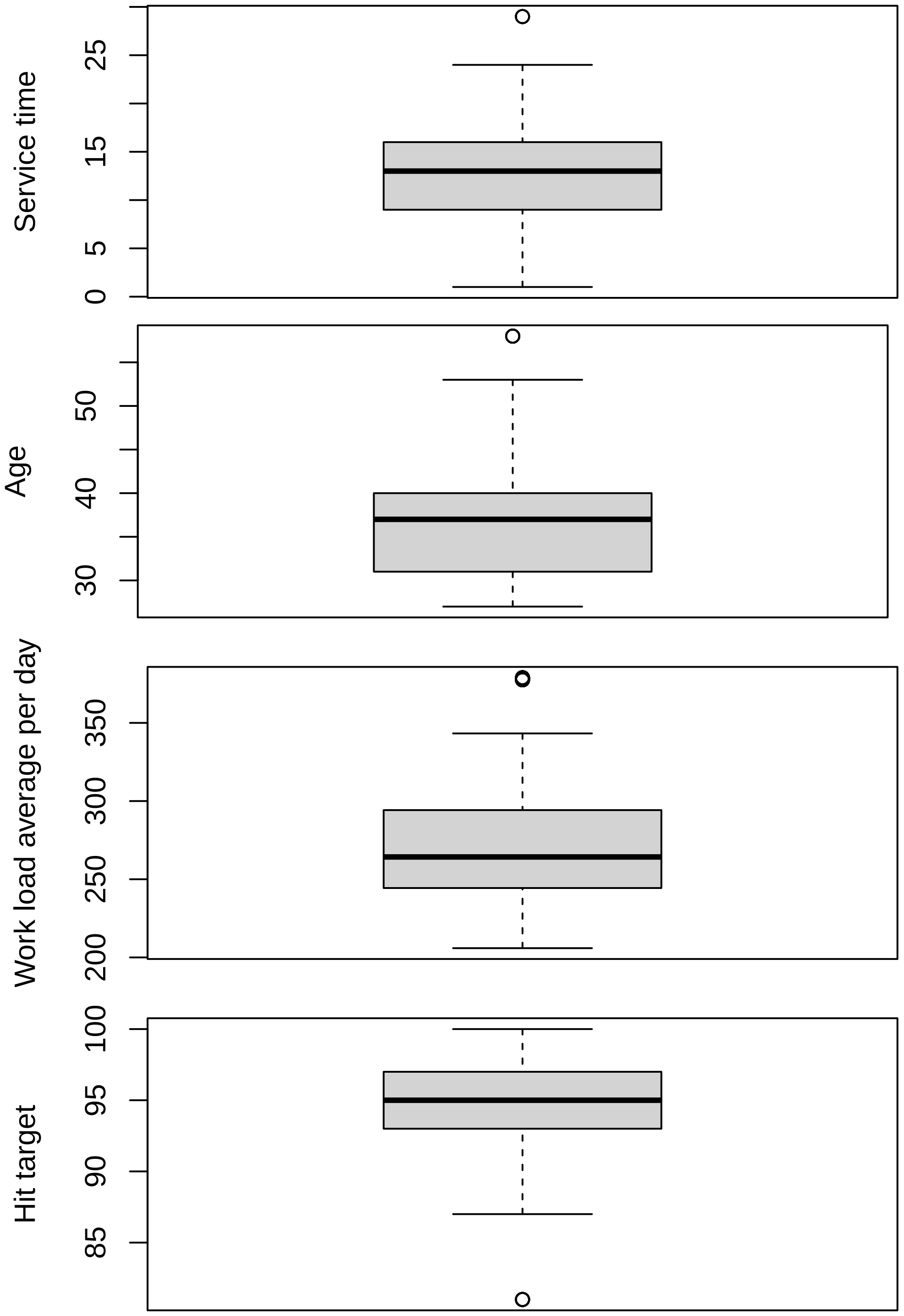
STATS/CSE 780
Assignment 3

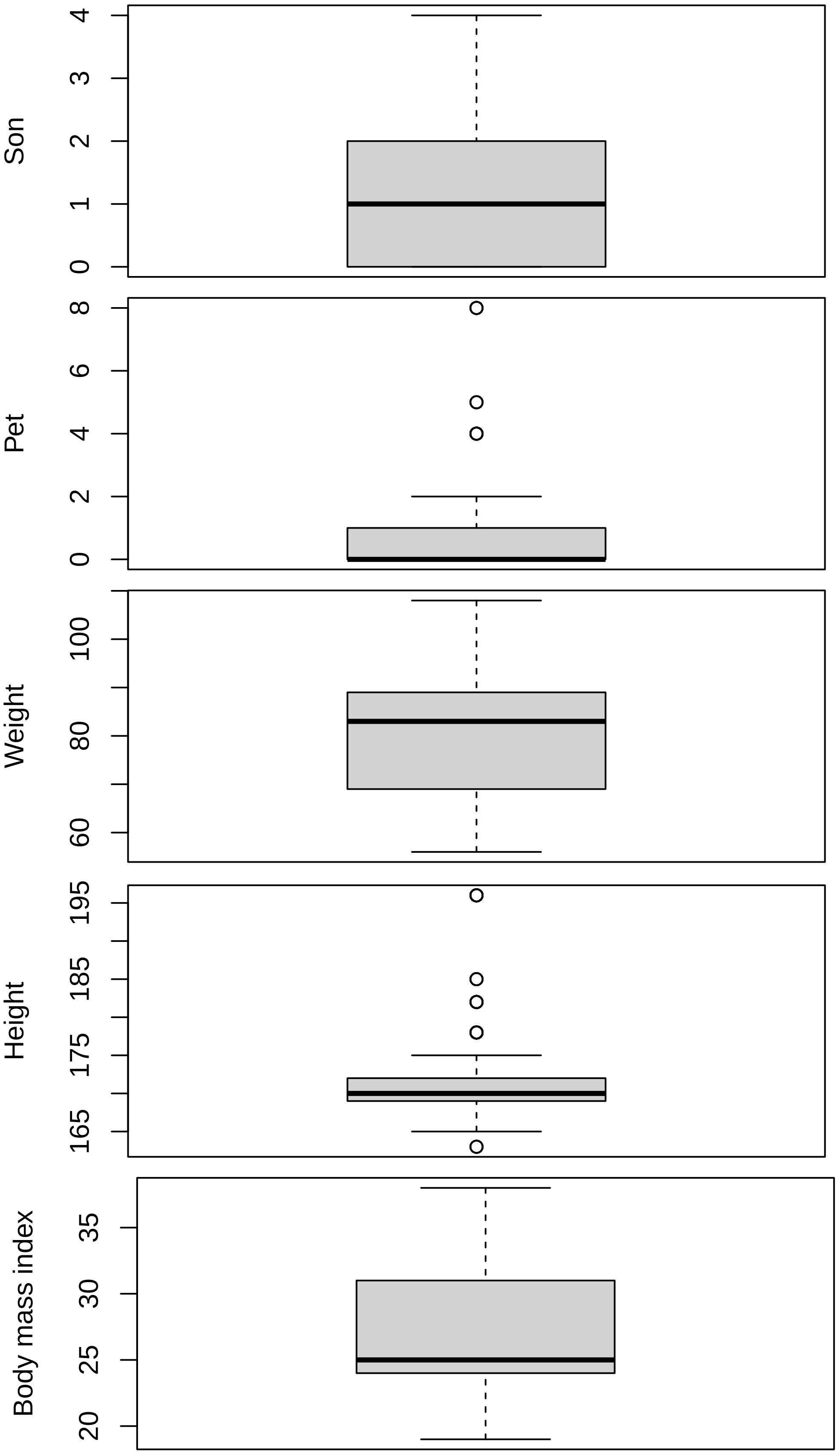
Pao Zhu Vivian Hsu (Student Number: 400547994)

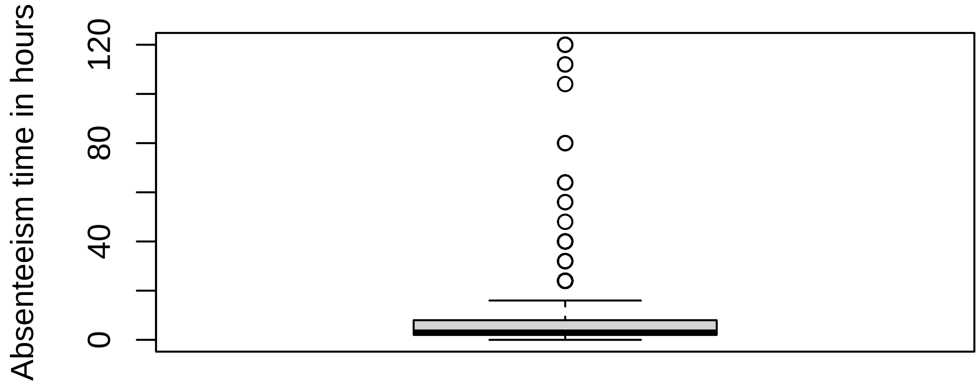
2023-11-07

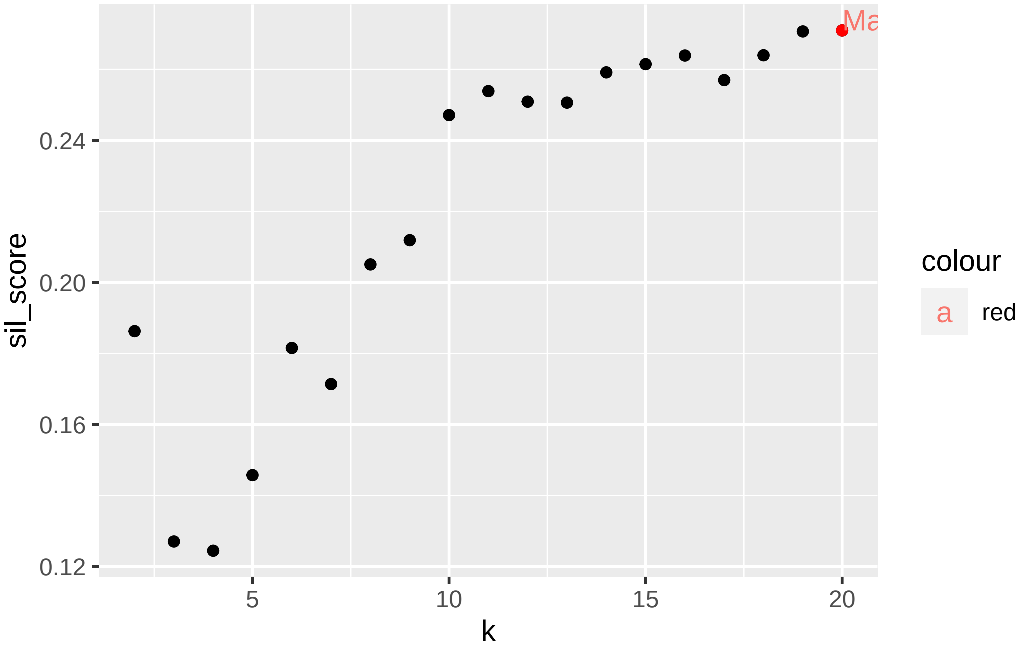
	data_type	min	max	nulls_blanks
Transportation.expense	integer	118.000	388.000	0
Distance.from.Residence.to.Work	integer	5.000	52.000	0
Service.time	integer	1.000	29.000	0
Age	integer	27.000	58.000	0
Work.load.Average.day	numeric	205.917	378.884	0
Hit.target	integer	81.000	100.000	0
Son	integer	0.000	4.000	0
Pet	integer	0.000	8.000	0
Weight	integer	56.000	108.000	0
Height	integer	163.000	196.000	0
Body.mass.index	integer	19.000	38.000	0
Absenteeism.time.in.hours	integer	0.000	120.000	0

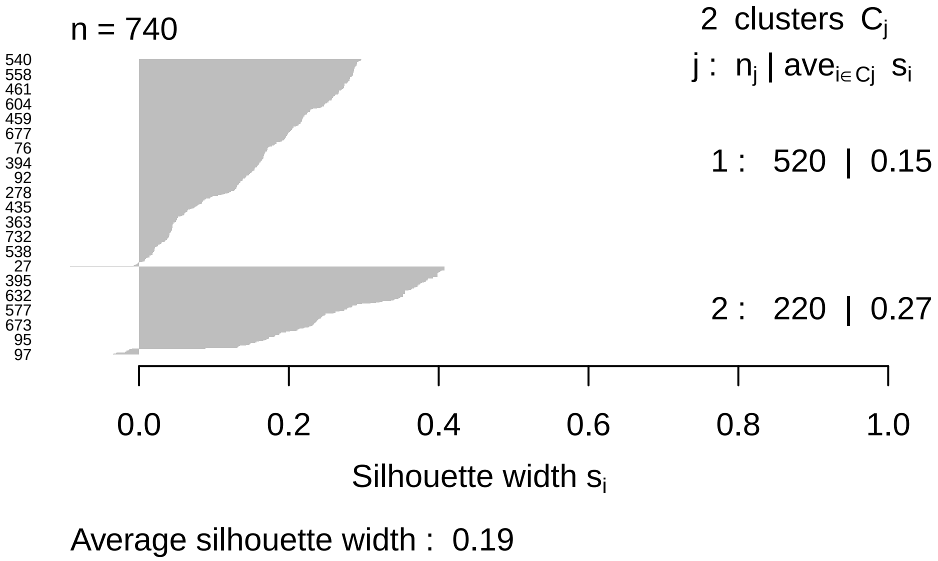



Service time
0
5
15
25
Age
30
40
50
Work load average per day
200
250
300
350
Hit target
85
90
95
100


Son
0
1
2
3
4
Pet
0
2
4
6
8
Weight
60
80
100
Height
165
175
185
195
Body mass index
20
25
30
35

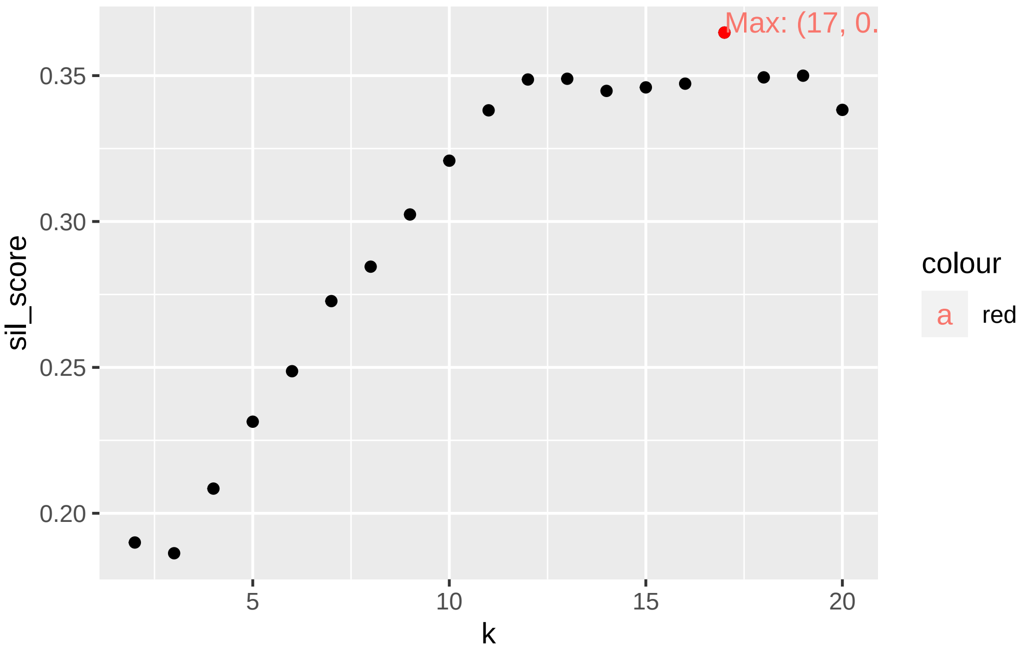

Absenteeism time in hours
0
40
80
120

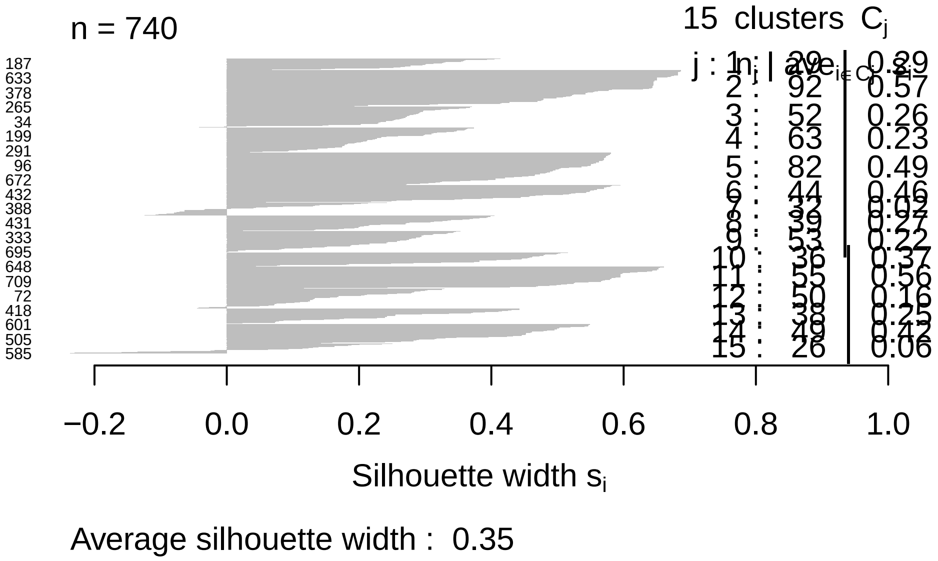

Ma
sil_score
0.12
0.16
0.20
0.24
5
10
15
20
k
colour
a red

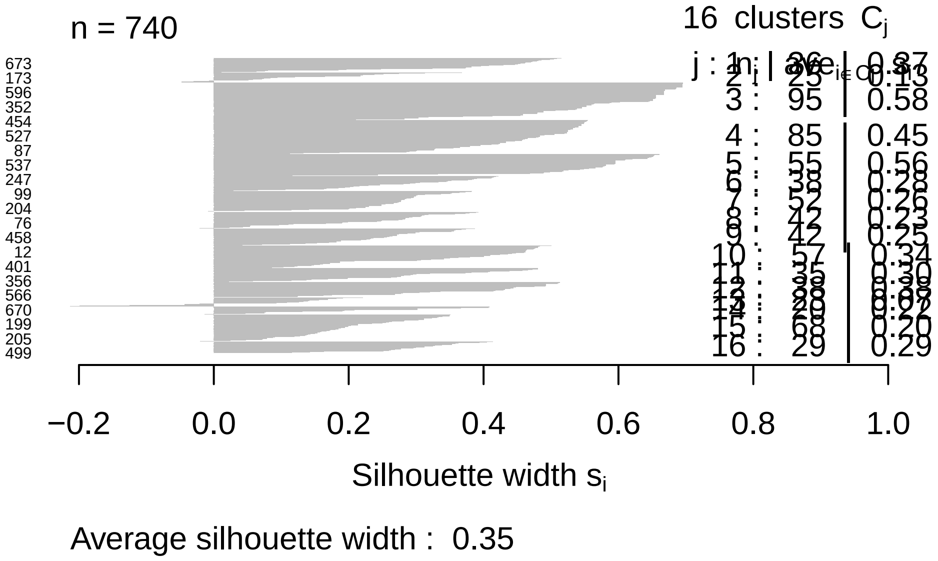

n = 740
2 clusters C_j
j : n_j | $ave_{i \in Cj}$ s_i
1 : 520 | 0.15
2 : 220 | 0.27
540
558
461
604
459
677
76
394
92
278
435
363
732
538
27
395
632
577
673
95
97
0.0
0.2
0.4
0.6
0.8
1.0
Silhouette width s_i
Average silhouette width : 0.19

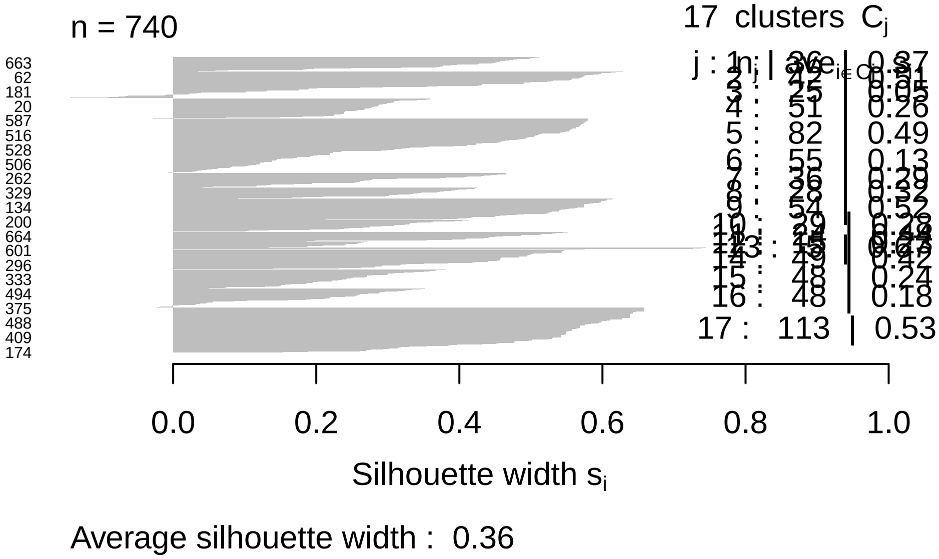
```
[1] 0.8410708
```

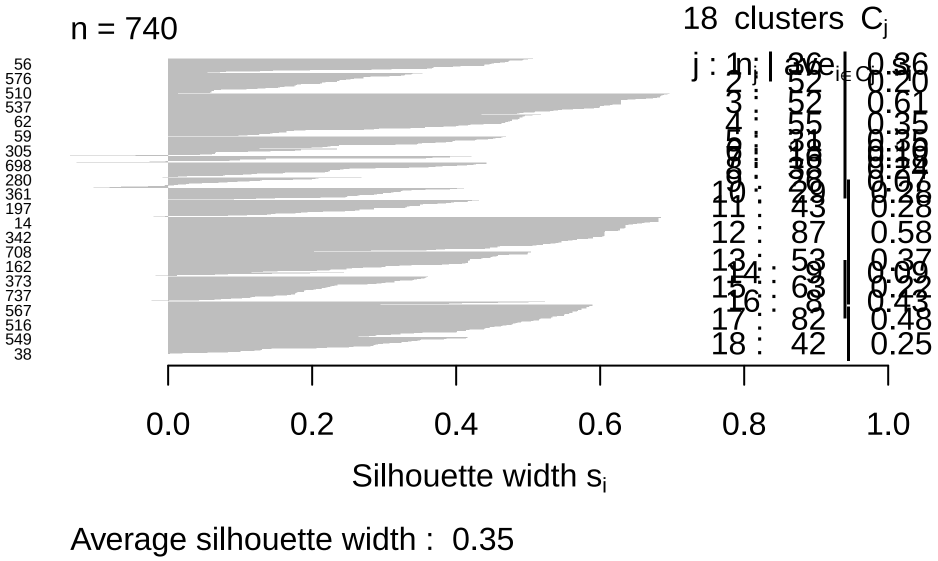
```
[1] 0.09340035
```

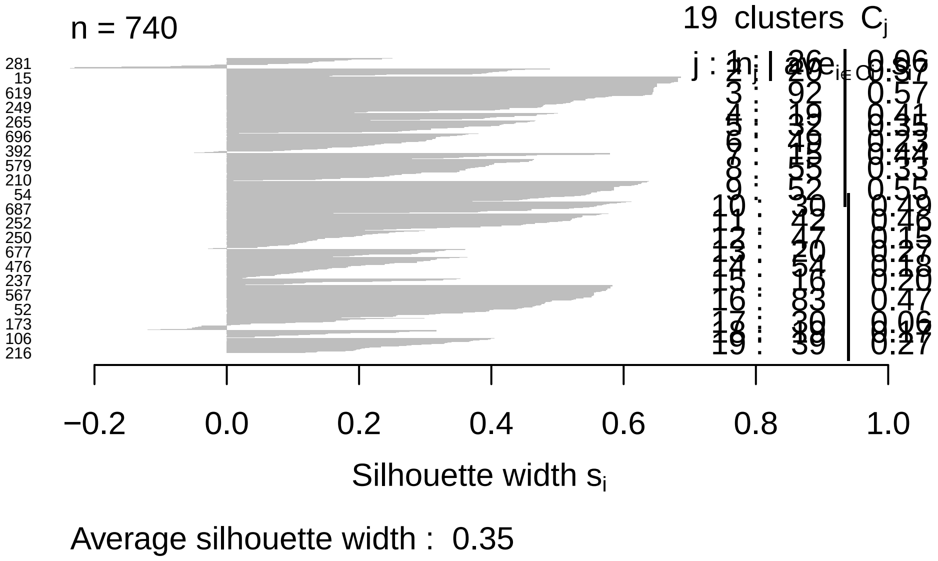

Max: (17, 0.
sil_score
0.35
0.30
0.25
0.20
5
10
15
20
k
colour
a
red

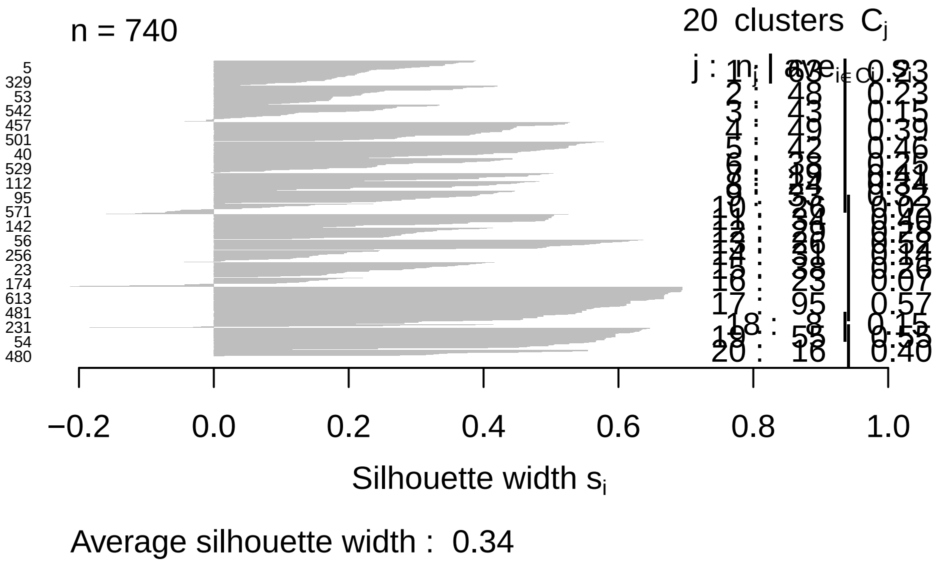

n = 740
15 clusters C_j
j : n_j | $\text{ave}_{i \in C_j}\ s_i$
1 : 29 | 0.29
2 : 92 | 0.57
3 : 52 | 0.26
4 : 63 | 0.23
5 : 82 | 0.49
6 : 44 | 0.46
7 : 32 | 0.02
8 : 39 | 0.27
9 : 53 | 0.22
10 : 36 | 0.37
11 : 55 | 0.56
12 : 50 | 0.16
13 : 38 | 0.25
14 : 49 | 0.42
15 : 26 | 0.06
187
633
378
265
34
199
291
96
672
432
388
431
333
695
648
709
72
418
601
505
585
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
Silhouette width s_i
Average silhouette width : 0.35


n = 740
16 clusters C_j
j : n_j | ave_{i∈Cj} s_i
3 : 95 | 0.58
4 : 85 | 0.45
5 : 55 | 0.56
9 : 42 | 0.25
16 : 29 | 0.29
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
Silhouette width s_i
Average silhouette width : 0.35


n = 740
17 clusters C_j
j : n_j | ave_{i∈Cj} s_i
4 : 51 | 0.26
5 : 82 | 0.49
6 : 55 | 0.13
15 : 48 | 0.24
16 : 48 | 0.18
17 : 113 | 0.53
0.0
0.2
0.4
0.6
0.8
1.0
Silhouette width s_i
Average silhouette width : 0.36

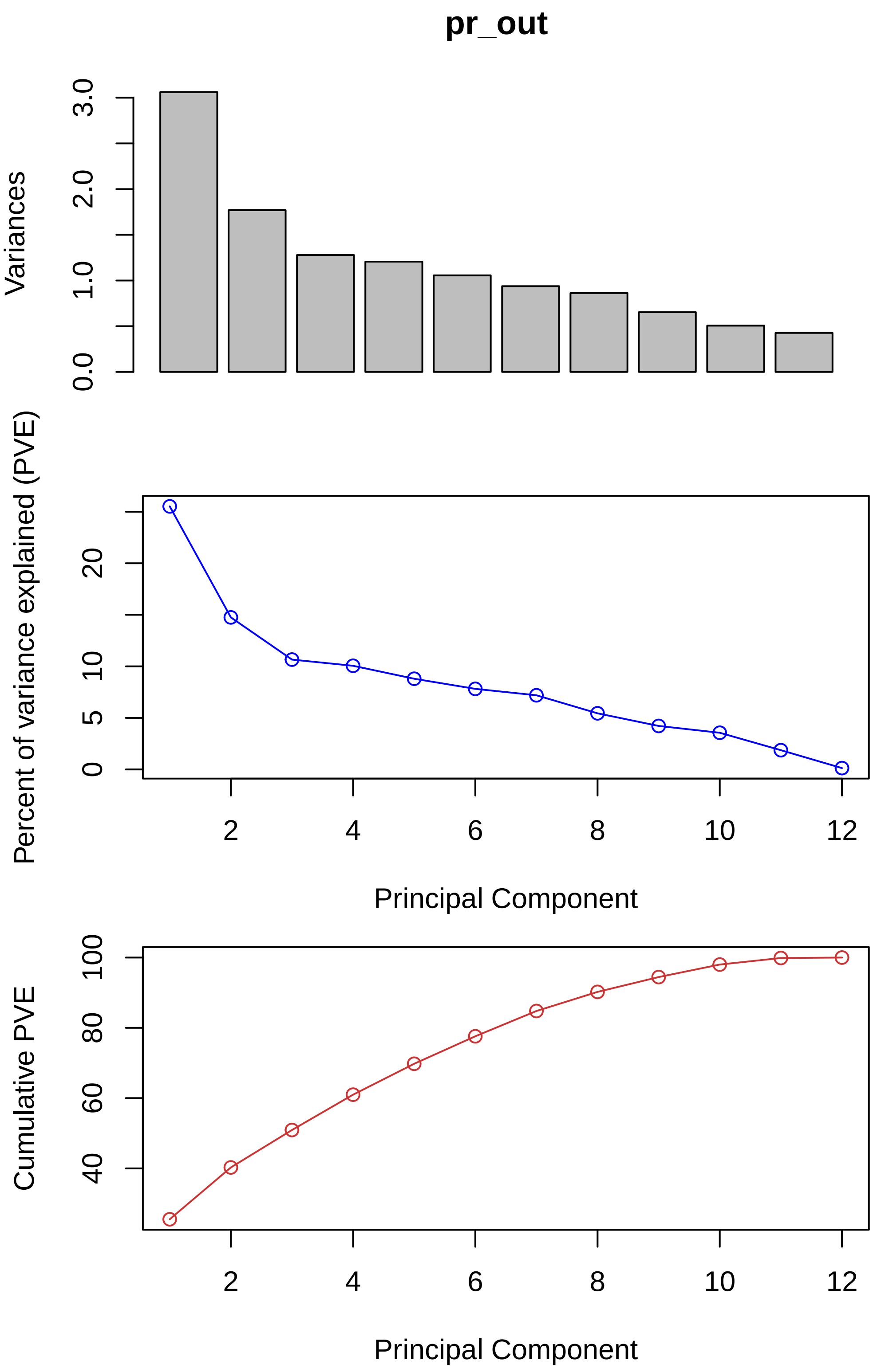

n = 740
18 clusters C_j
j : n_j | ave_{i∈Cj} s_i
3 : 52 | 0.61
4 : 55 | 0.35
11 : 43 | 0.28
12 : 87 | 0.58
17 : 82 | 0.48
18 : 42 | 0.25
0.0
0.2
0.4
0.6
0.8
1.0
Silhouette width s_i
Average silhouette width : 0.35

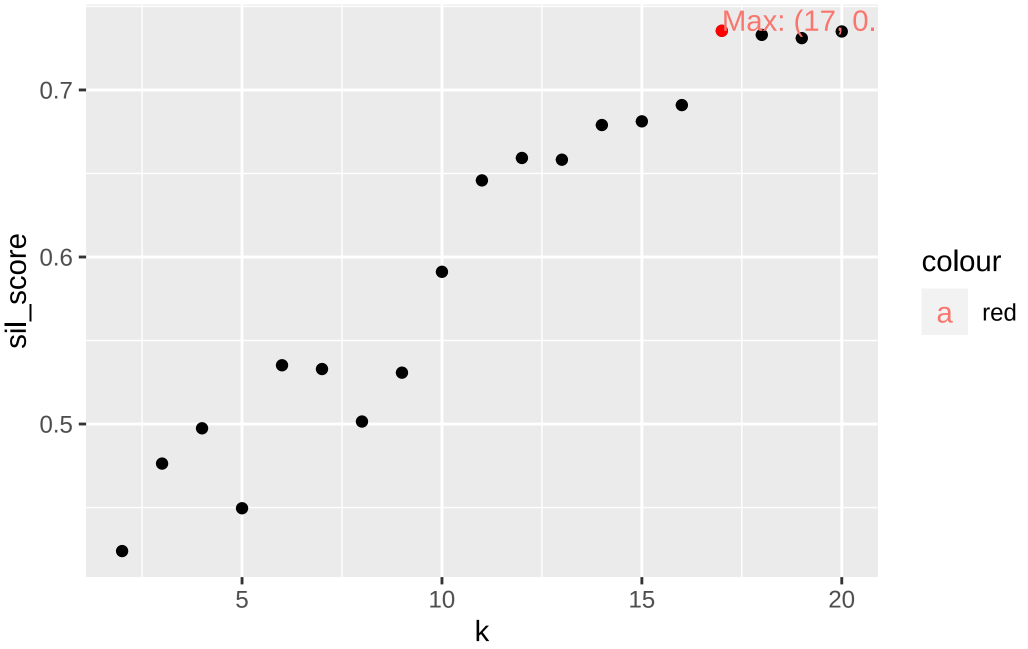


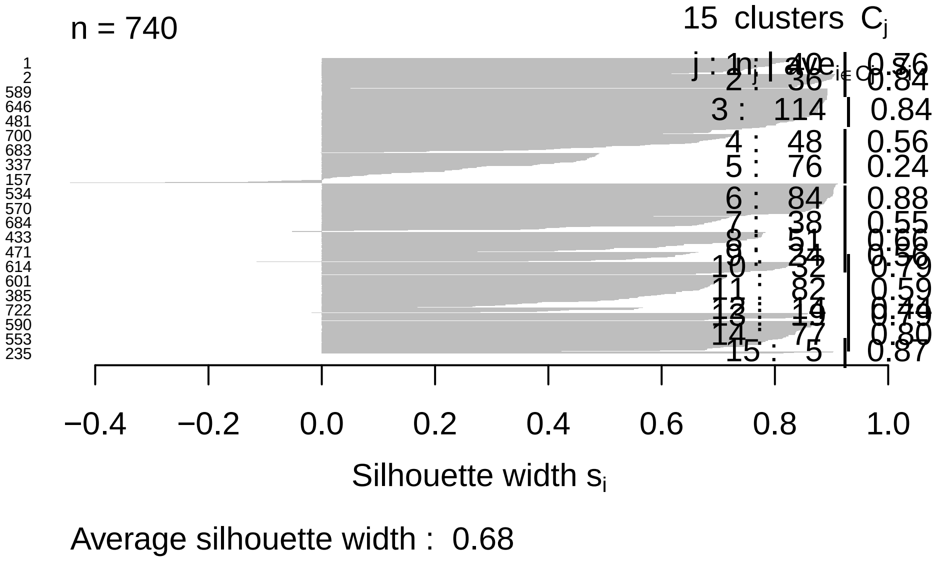


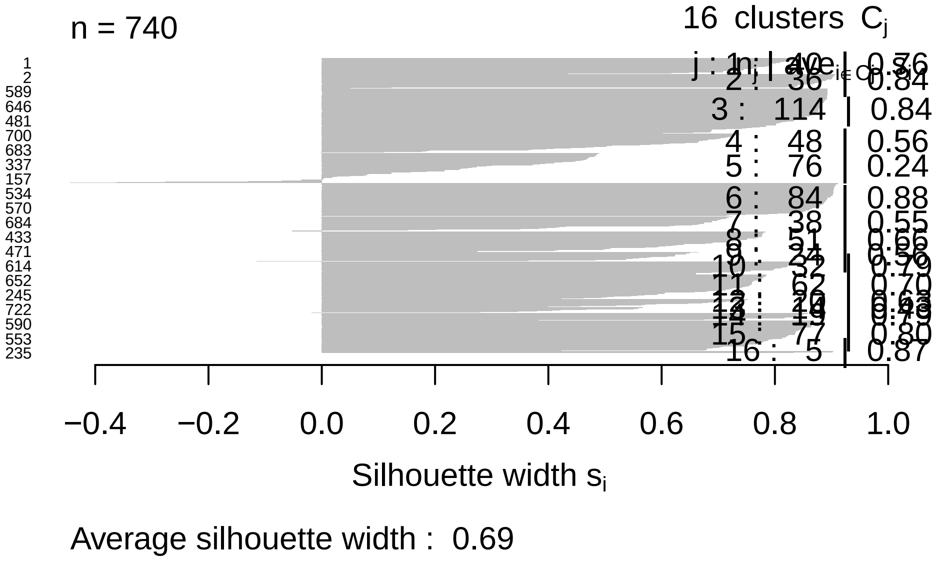
```
[1] 0.8539041
```

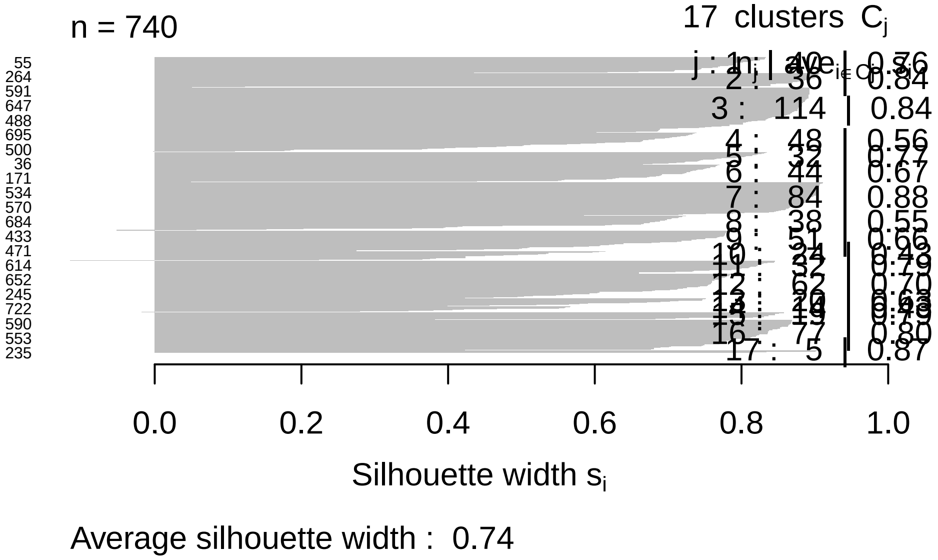
```
[1] 0.1017307
```

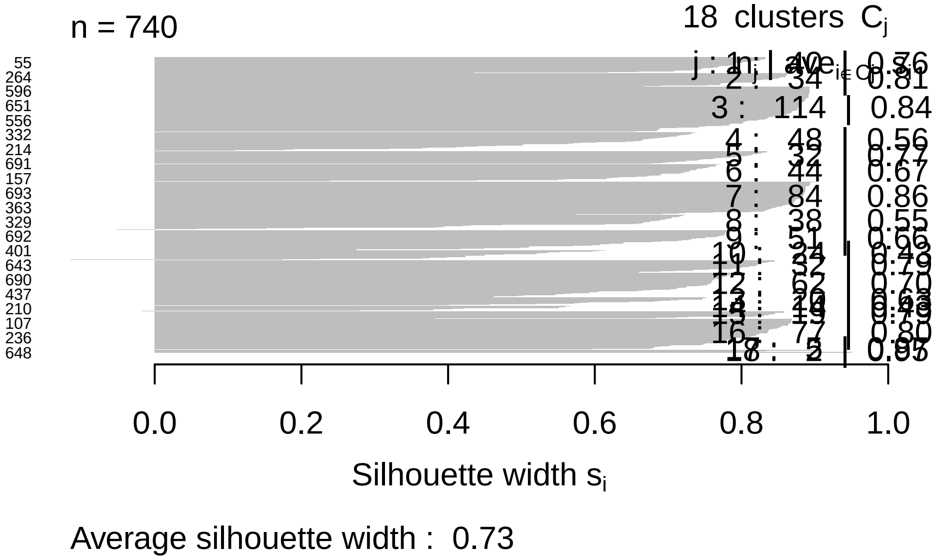

pr_out
Variances
0.0
1.0
2.0
3.0
Percent of variance explained (PVE)
0
5
10
20
2
4
6
8
10
12
Principal Component
Cumulative PVE
40
60
80
100
2
4
6
8
10
12
Principal Component

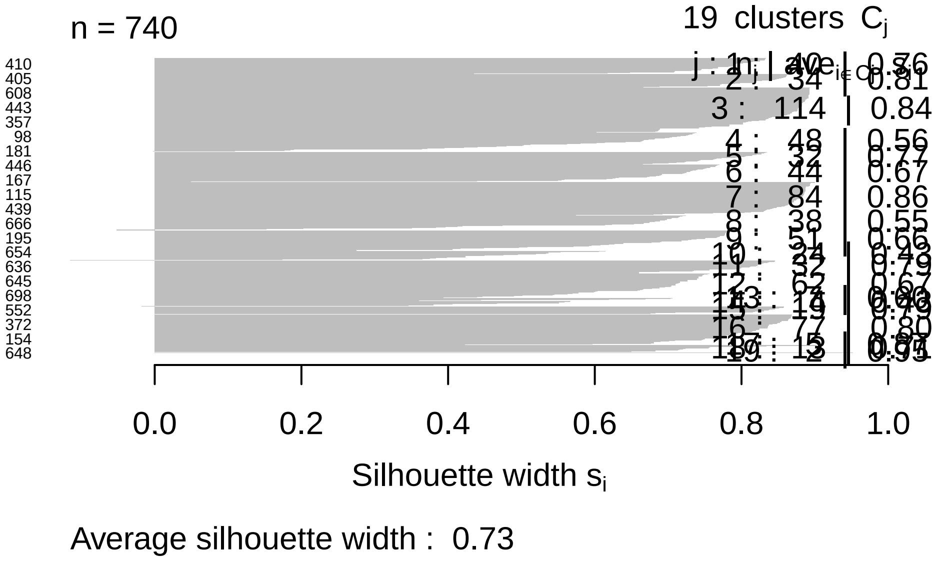

Max: (17, 0.
0.7
0.6
0.5
sil_score
5
10
15
20
k
colour
a red

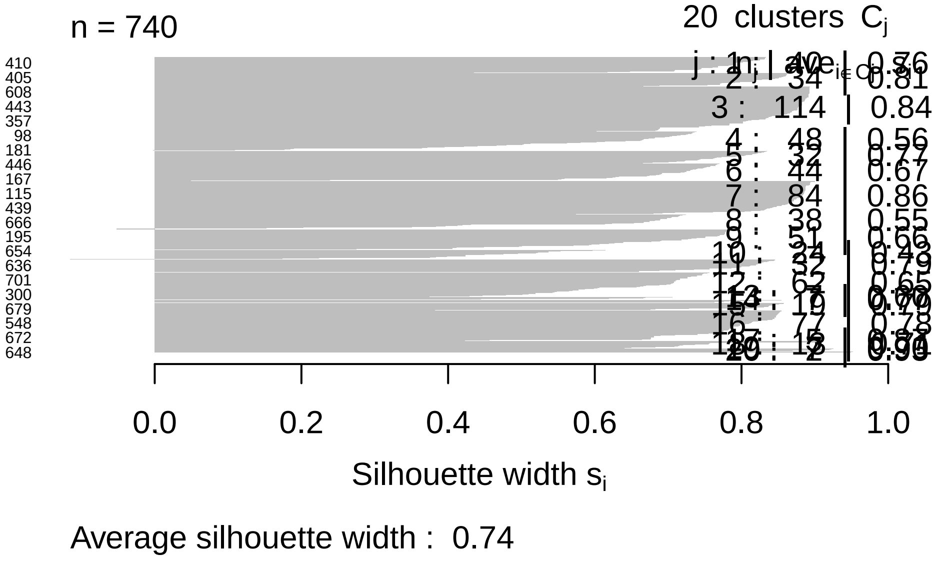

n = 740
15 clusters C_j
j : n_j | $ave_{i \in Cj}$ s_i
1 : 40 | 0.76
2 : 36 | 0.84
3 : 114 | 0.84
4 : 48 | 0.56
5 : 76 | 0.24
6 : 84 | 0.88
7 : 38 | 0.55
8 : 51 | 0.66
9 : 24 | 0.56
10 : 32 | 0.79
11 : 82 | 0.59
12 : 14 | 0.74
13 : 19 | 0.79
14 : 77 | 0.80
15 : 5 | 0.87
1
2
589
646
481
700
683
337
157
534
570
684
433
471
614
601
385
722
590
553
235
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
Silhouette width s_i
Average silhouette width : 0.68


n = 740
16 clusters C_j
j : n_j | ave_{i∈Cj} s_i
1 : 40 | 0.76
2 : 36 | 0.84
3 : 114 | 0.84
4 : 48 | 0.56
5 : 76 | 0.24
6 : 84 | 0.88
7 : 38 | 0.55
8 : 51 | 0.66
15 : 77 | 0.80
16 : 5 | 0.87
1
2
589
646
481
700
683
337
157
534
570
684
433
471
614
652
245
722
590
553
235
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
Silhouette width s_i
Average silhouette width : 0.69


n = 740
17 clusters C_j
j : n_j | ave_{i∈Cj} s_i
1 : 40 | 0.76
2 : 36 | 0.84
3 : 114 | 0.84
4 : 48 | 0.56
6 : 44 | 0.67
7 : 84 | 0.88
8 : 38 | 0.55
9 : 51 | 0.66
12 : 62 | 0.70
16 : 77 | 0.80
17 : 5 | 0.87
55
264
591
647
488
695
500
36
171
534
570
684
433
471
614
652
245
722
590
553
235
0.0
0.2
0.4
0.6
0.8
1.0
Silhouette width s_i
Average silhouette width : 0.74


n = 740
18 clusters C_j
j : n_j | ave_{i∈Cj} s_i
1 : 40 | 0.76
2 : 34 | 0.81
3 : 114 | 0.84
4 : 48 | 0.56
6 : 44 | 0.67
7 : 84 | 0.86
8 : 38 | 0.55
12 : 62 | 0.70
16 : 77 | 0.80
55
264
596
651
556
332
214
691
157
693
363
329
692
401
643
690
437
210
107
236
648
0.0
0.2
0.4
0.6
0.8
1.0
Silhouette width s_i
Average silhouette width : 0.73





```
[1] 0.8454778
```

```
[1] 0.1130238
```

Introduction

Methods

Discussion

Supplementary material

```
# ----- SETUP ----- #
# Load packages
packages <- c("knitr", "tidyverse", "ggplot2", "cluster", "fossil")
lapply(packages, library, character.only = TRUE)

# Read data, extract labels, and keep only quantitative data
absentData_raw <- read.csv("Absenteeism_at_work.csv", sep = ";")
absentData_lab <- absentData_raw$`Reason.for.absence`
absentData <- absentData_raw %>%
  select(-c("Reason.for.absence","ID","Month.of.absence","Day.of.the.week","Seasons",
            "Disciplinary.failure","Education","Social.drinker","Social.smoker"))
# ----- DATA EXPLORATION ----- #
# Check data types, min, max, and missing data
data_type <- sapply(absentData,class)
min <- sapply(absentData, function(col){min(col,na.rm=TRUE)})
max <- sapply(absentData, function(col){max(col,na.rm=TRUE)})
nulls <- sapply(absentData, function(col){sum(is.na(col))})
blanks <- sapply(absentData,
                 function(col){ifelse(is.na(sum(col == "")), 0, sum(col == ""))})
data_summary <- data.frame(row.names = names(nulls), data_type=data_type,
                           min=min, max=max, nulls_blanks=nulls+blanks)
kable(data_summary)

# Create box plots to check for outliers
b01 <- boxplot(absentData$Transportation.expense, ylab = "Transportation Expense")
b02 <- boxplot(absentData$Distance.from.Residence.to.Work,
               ylab = "Distance from residence to work")
b03 <- boxplot(absentData$Service.time, ylab = "Service time")
b04 <- boxplot(absentData$Age, ylab = "Age")
b05 <- boxplot(absentData$Work.load.Average.day, ylab = "Work load average per day")
```

```
b06 <- boxplot(absentData$Hit.target, ylab = "Hit target")
b07 <- boxplot(absentData$Son, ylab = "Son")
b08 <- boxplot(absentData$Pet, ylab = "Pet")
b09 <- boxplot(absentData$Weight, ylab = "Weight")
b10 <- boxplot(absentData$Height, ylab = "Height")
b11 <- boxplot(absentData$Body.mass.index, ylab = "Body mass index")
b12 <- boxplot(absentData$Absenteeism.time.in.hours, ylab = "Absenteeism time in hours")

# ----- DATA CLEANSING -----
# Handle outliers by capping them using interquartile range
cap <- function(val, bplot) {
  lower_fence <- bplot$stats[2]-(1.5*(bplot$stats[4]-bplot$stats[2])) #Q1-1.5*IQR
  upper_fence <- bplot$stats[4]+(1.5*(bplot$stats[4]-bplot$stats[2])) #Q3+1.5*IQR
  val <- ifelse(val < lower_fence, lower_fence, val)
  val <- ifelse(val > upper_fence, upper_fence, val)
  val
}
absentData <- absentData %>%
  mutate(Transportation.expense = cap(val=Transportation.expense, bplot=b01),
         Service.time = cap(val=Service.time, bplot=b03),
         Age = cap(val=Age, bplot=b04),
         Work.load.Average.day = cap(val=Work.load.Average.day, bplot=b05),
         Hit.target = cap(val=Hit.target, bplot=b06),
         Pet = cap(val=Pet, bplot=b08),
         Height = cap(val=Height, bplot=b10),
         Absenteeism.time.in.hours = cap(val=Absenteeism.time.in.hours, bplot=b12))

# Scale the data
absentData_sd <- scale(absentData)

# ----- CLUSTERING FUNCTIONS ----- #
```

```r
# Get silhouette for k-means clustering
kmcSilK <- function(k, data){
  x_k <- kmeans(data, k, nstart = 20)
  silhouette(x_k$cluster, dist(data))
}

# Get silhouette for hierarchical clustering
hcSilK <- function(k, data){
  hc_out <- hclust(dist(data))
  hc_clusters <- cutree(hc_out, k)
  silhouette(hc_clusters, dist(data))
}

# Plot silhouette
plotSil <- function(sil){
  plot(sil, nmax= 800, cex.names=0.5, main = "", border=NA)
}

# Choose k using goodness-of-clustering
# k = the k values to test
# silFun = the silhouette function
# data = the data used in silFun
chooseK <- function(k, silFun, data) {

  # Get silhouettes and their widths
  sil_k <- lapply(k, silFun, data=data)
  sil_score <- sapply(sil_k, function(x) {mean(x[,"sil_width"])})

  # Find the k with the max width
  sil_max <- max(sil_score)
  sil_max_k <- match(sil_max, sil_score)+min(k)-1
  # Plot the silhouette widths and label the maximum
  silData <- tibble(k, sil_score)
  max_point <-tibble(k=sil_max_k,sil_score=sil_max)
  max_lab <- paste0("Max: (",sil_max_k,", ",round(sil_max,2), ")")
  plot <- ggplot(silData, aes(x=k, y=sil_score)) + geom_point() +
    geom_point(data=max_point, colour="red") +
    geom_text(data=max_point, aes(label=ifelse(k==sil_max_k,max_lab,""), color="red"),hjust=

  # Return plot, silhouettes, and max k
  list(plot=plot, sil_k=sil_k, max_k=sil_max_k)
}

#plotSil(hcSilK(2, absentData_sd)) # hierarchical clustering
#plotSil(hcSilK(2, pr_out$x[, 1:2])) # hierarchical clustering after PCA
#chooseK(2:30, hcSilK, absentData_sd)
#chooseK(2:30, hcSilK, pr_out$x[, 1:2])

# ----- AGGLOMERATIVE HIERARCHICAL CLUSTERING ----- #
# Get silhouette scores for multiple k values and plot them
set.seed(3)
k <- c(2:20)
good_of_cluster <- chooseK(k, hcSilK, absentData_sd) # uses complete linkage
good_of_cluster$plot
plotSil(good_of_cluster$sil_k[[1]])
k <- good_of_cluster$max_k

# Perform hierarchical clustering using best k
set.seed(3)
hc_out <- hclust(dist(absentData_sd))
rand.index(cutree(hc_out, k), as.numeric(as.factor(absentData_lab)))
```

```r
adj.rand.index(cutree(hc_out, k), as.numeric(as.factor(absentData_lab)))

# ----- K-MEANS CLUSTERING ----- #
# Get silhouette scores for multiple k values and plot them
set.seed(3)
k <- c(2:20)
good_of_cluster <- chooseK(k, kmcSilK, absentData_sd)
good_of_cluster$plot

# Based off silhouette plots, choose the best k
plotSil(good_of_cluster$sil_k[[14]])
plotSil(good_of_cluster$sil_k[[15]])
plotSil(good_of_cluster$sil_k[[16]])
plotSil(good_of_cluster$sil_k[[17]])
plotSil(good_of_cluster$sil_k[[18]])
plotSil(good_of_cluster$sil_k[[19]])
k <- 16

# Perform k-means clustering with best k and compute the rand indices
set.seed(3)
km_out <- kmeans(absentData, k, nstart = 20)
km_clusters <- km_out$cluster
rand.index(km_clusters, as.numeric(as.factor(absentData_lab)))
adj.rand.index(km_clusters, as.numeric(as.factor(absentData_lab)))

# ----- HIERARCHICAL CLUSTERING AFTER PCA ----- #
# Proportion of variance explained
set.seed(3)
pr_out <- prcomp(absentData, scale = TRUE)
plot(pr_out)
```

```
pve <- 100 * pr_out$sdev^2 / sum(pr_out$sdev^2)
plot(pve, type = "o",
xlab = "Principal Component", col = "blue", ylab = "Percent of variance explained (PVE)")

plot(cumsum(pve), type = "o", ylab = "Cumulative PVE",
     xlab = "Principal Component", col = "brown3")

# Get silhouette scores for multiple k values and plot them
set.seed(3)
k <- c(2:20)
good_of_cluster <- chooseK(k, hcSilK, pr_out$x[, 1:2])
good_of_cluster$plot

# Based off silhouette plots, choose the best k
plotSil(good_of_cluster$sil_k[[14]])
plotSil(good_of_cluster$sil_k[[15]])
plotSil(good_of_cluster$sil_k[[16]])
plotSil(good_of_cluster$sil_k[[17]])
plotSil(good_of_cluster$sil_k[[18]])
plotSil(good_of_cluster$sil_k[[19]])
k <- good_of_cluster$max_k

set.seed(3)
hc_out <- hclust(dist(dist(pr_out$x[, 1:2])))
hc_clusters <- cutree(hc_out, k)
rand.index(hc_clusters, as.numeric(as.factor(absentData_lab)))
adj.rand.index(hc_clusters, as.numeric(as.factor(absentData_lab)))
```

References

R Core Team. (2023). *R: A language and environment for statistical computing.* R Foundation for Statistical Computing. https://www.R-project.org/