STATS/CSE 780
Assignment 3

Pao Zhu Vivian Hsu (Student Number: 400547994)

2023-11-09

Introduction

Companies rely on employees to carry out operations and achieve business goals. However, when employees are regularly absent from work, carrying out business can become expensive in terms of both finances and time (Araujo et al., 2019). In this study, we apply clustering methods to understand and predict the underlying reasons behind absenteeism at work. The results will provide companies with direction on how to address these reasons and ultimately reduce rates of employee absenteeism.

Methods

The data in this study was sourced from the UC Irvine Machine Learning Repository (Martiniano & Ferreira, 2018). It contains 21 variables and 740 records of absenteeism at work from 2007 to 2010 at a courier company in Brazil (Martiniano & Ferreira, 2018). The data set includes a categorical variable indicating the reason for absenteeism and various variables describing the employee and their working conditions. The categorical variable for absenteeism combined with quantitative variables makes this a suitable data set for hierarchical and k-means clustering. Since we are only interested in clustering using quantitative variables, all categorical variables apart from the reason for absenteeism were removed from the data set, resulting in a total of 12 variables.

Prior to clustering, we first explored the data to check if data transformation was required. There were no missing values, unexpected data types, or unreasonable data types detected in the data. Of the 12 variables, 8 of them had outliers as shown in the box plots in Supp. Materials Figure 3. These were capped by the lower and upper fences of the interquartile range. We also scaled and centered the data.

Once data transformation was done, we then applied clustering methods to the data. We first performed hierarchical clustering. Silhouette plots for cluster sizes 2 to 20 were created and the average silhouette widths were plotted as shown in the left plot in Figure 1. A cluster size of 20 resulted in the maximum average silhouette width, however we decided to use 10 clusters instead of 20 because the average widths are quite similar for a cluster size of 20 versus 10 as shown in Figure 1, and there are many negative silhouette widths (i.e. samples placed into the wrong cluster) for a cluster size of 20 as shown in Figure 5.

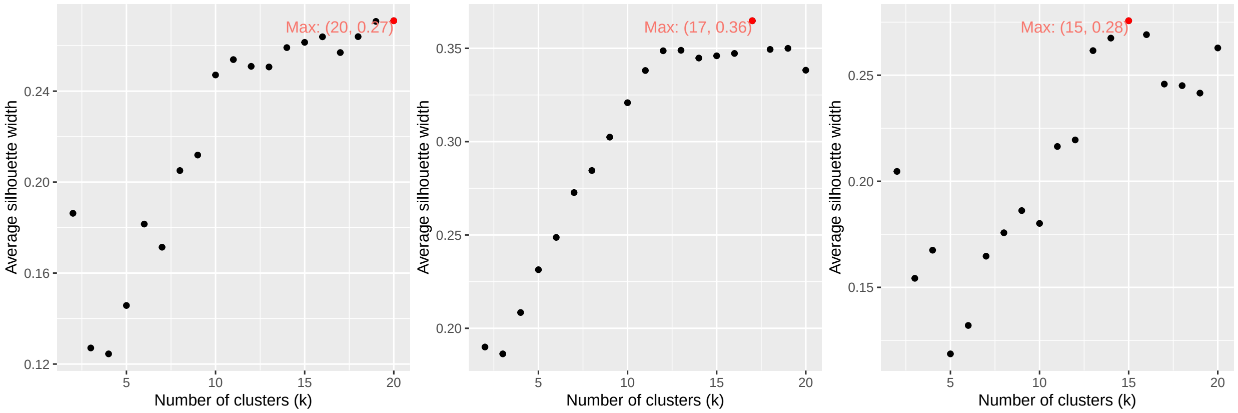

Figure 1: Average silhouette widths for cluster sizes 1 to 20; Left chart is for hierarchical clustering, centre chart is for k-means clustering, and right chart is for PCA followed by hierarchical clustering.

Next, we performed k-means clustering. Similar to the previous method, we created silhouette plots for cluster sizes 2 to 20 and looked for the maximum average silhouette width. In this case, a cluster size of 17 produced the highest average as displayed in the centre plot of Figure 1. We decided to use 17 clusters since other cluster sizes did not show strong improvements in terms of negative silhouette widths as shown in Supp. Materials Figure 5.

Finally, we performed principal component analysis (PCA) followed by hierarchical clustering. 8 principal components (PCs) and 14 clusters were used. As shown in Figure 2, we decided the PCs using a scree plot and selected the number of components where approximately 85% of the variation could be explained. Similar to the previous two methods, we then created silhouette plots for cluster sizes of 2 to 20. We selected the cluster size with the maximum average silhouette width as illustrated in the right plot of Figure 1. We ensured that the negative silhouette width was minimal or comparable to other cluster sizes that produced a similar average silhouette width. This is shown in Supp. Materials Figure 6.

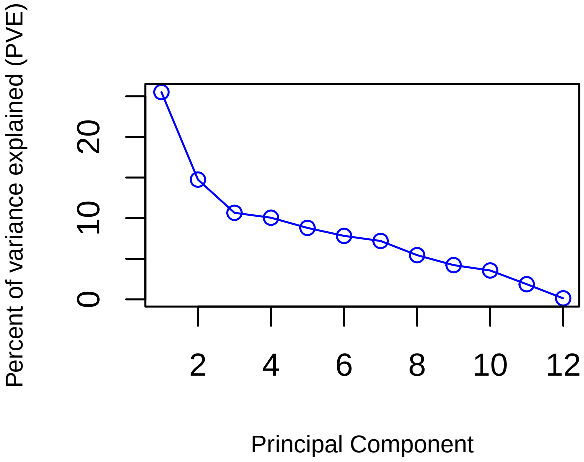

Figure 2: Scree plot

Results

Table 1 shows the results for the three clustering methods using Rand index and adjusted Rand index. All three methods produced a relatively high Rand index between 0.81 and 0.86. K-means clustering had the highest Rand index, followed closely by hierarchical clustering after PCA, and finally hierarchical clustering. Based off the Rand indices alone, it may appear that the clustering models accurately describe and predict absenteeism. However, their adjusted Rand indices indicate that this may not be the case. All three adjusted Rand indices were poor, ranging from 0.09 to 0.18. Hierarchical clustering had the highest adjusted Rand index, followed by hierarchical clustering after PCA, and finally k-means clustering. Overall, hierarchical clustering after PCA resulted in the most balanced results in terms of Rand index and adjusted Rand index, but the poor adjusted Rand index suggests that the clustering results may not be reliable. Further work is required to determine if another method, such as PCA followed by k-means clustering, would better model and predict employee absenteeism.

Table 1: Comparison of clustering methods using Rand index and adjusted Rand index

clustering_method	rand_index	adj_rand_index
hierarchical	0.812	0.176
k-means	0.853	0.092
pca followed by hierarchical	0.847	0.111

Supplementary material

Tables and figures

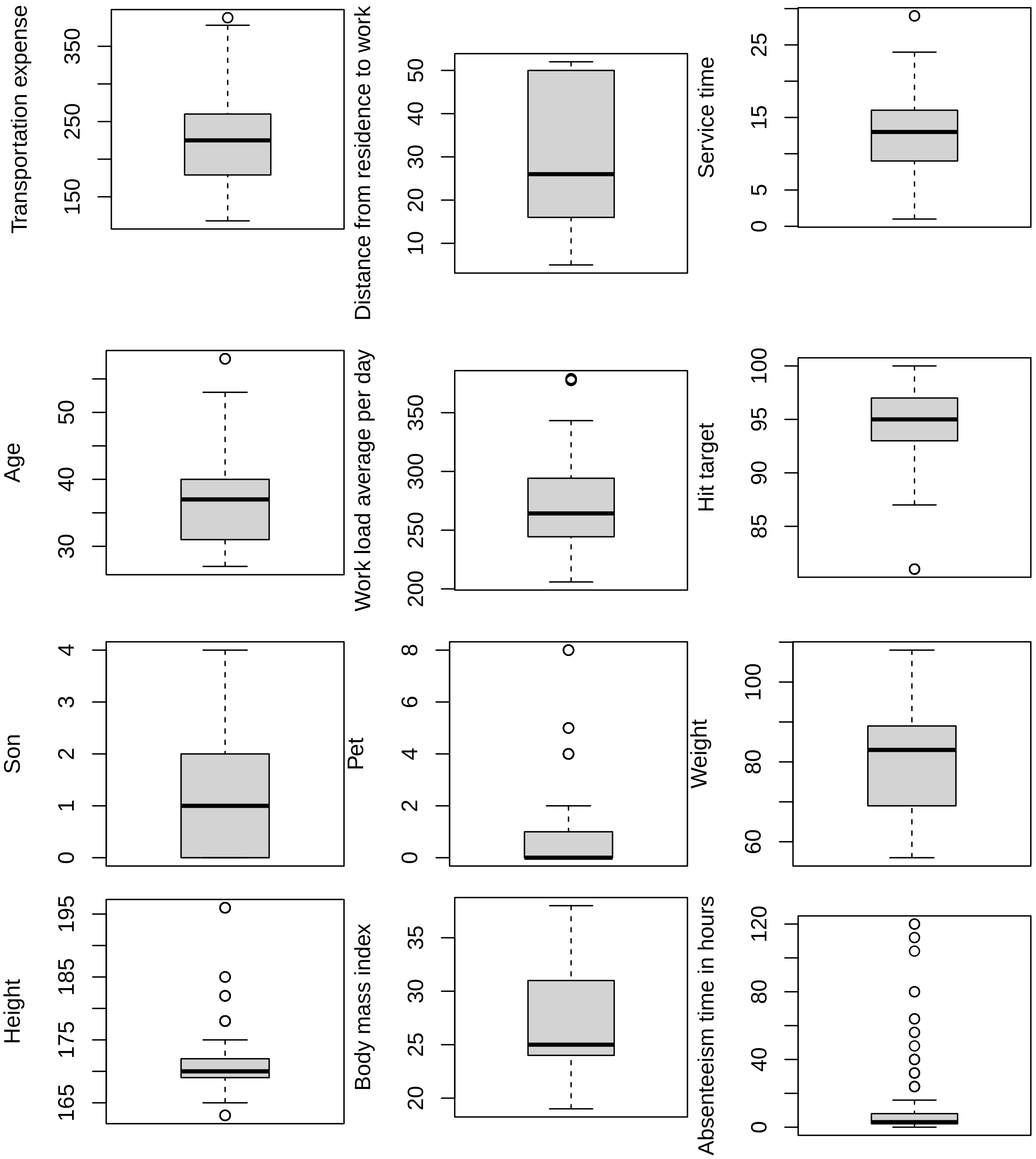

Figure 3: Outliers were detected from the box plots of some variables

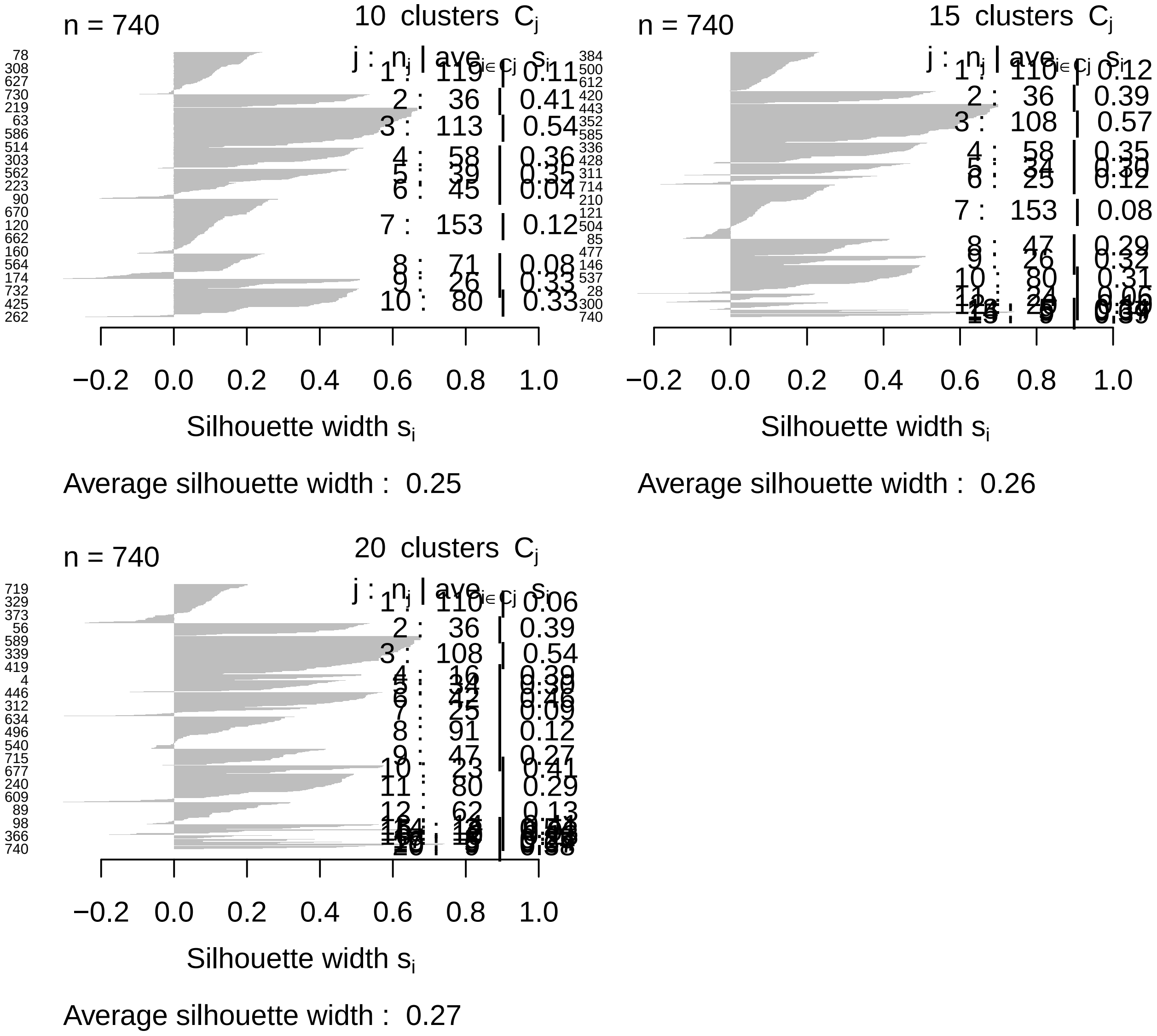

Figure 4: Silhouette plots for cluster sizes for hierarchical clustering. k = 20 returned the maximum silhouette width values while k = 10 and 15 are cluster sizes with similar silhouette widths.

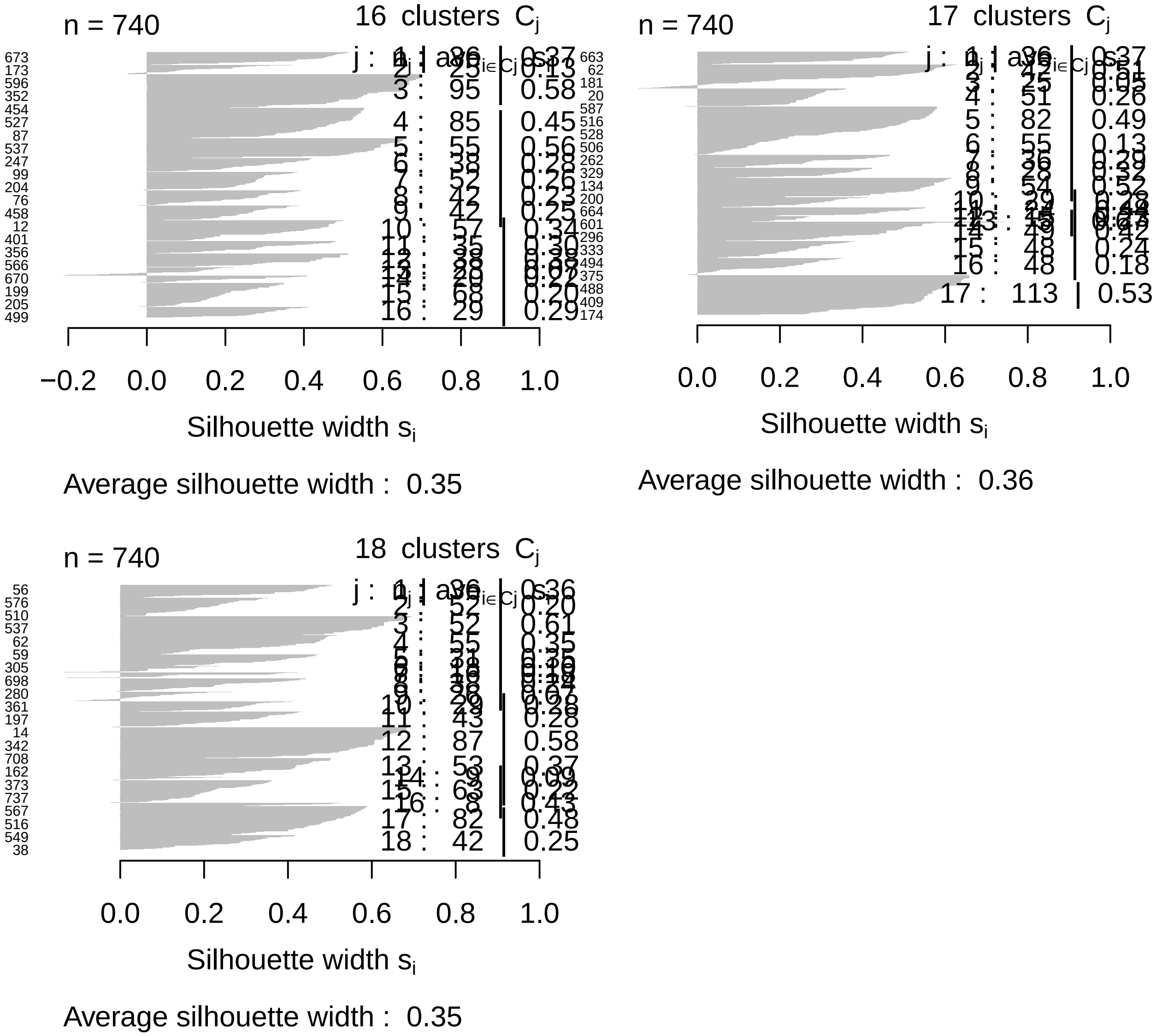

Figure 5: Silhouette plots for k-means clustering. k = 17 returned the maximum silhouette width values while k = 16 and 18 are cluster sizes with similar silhouette widths.

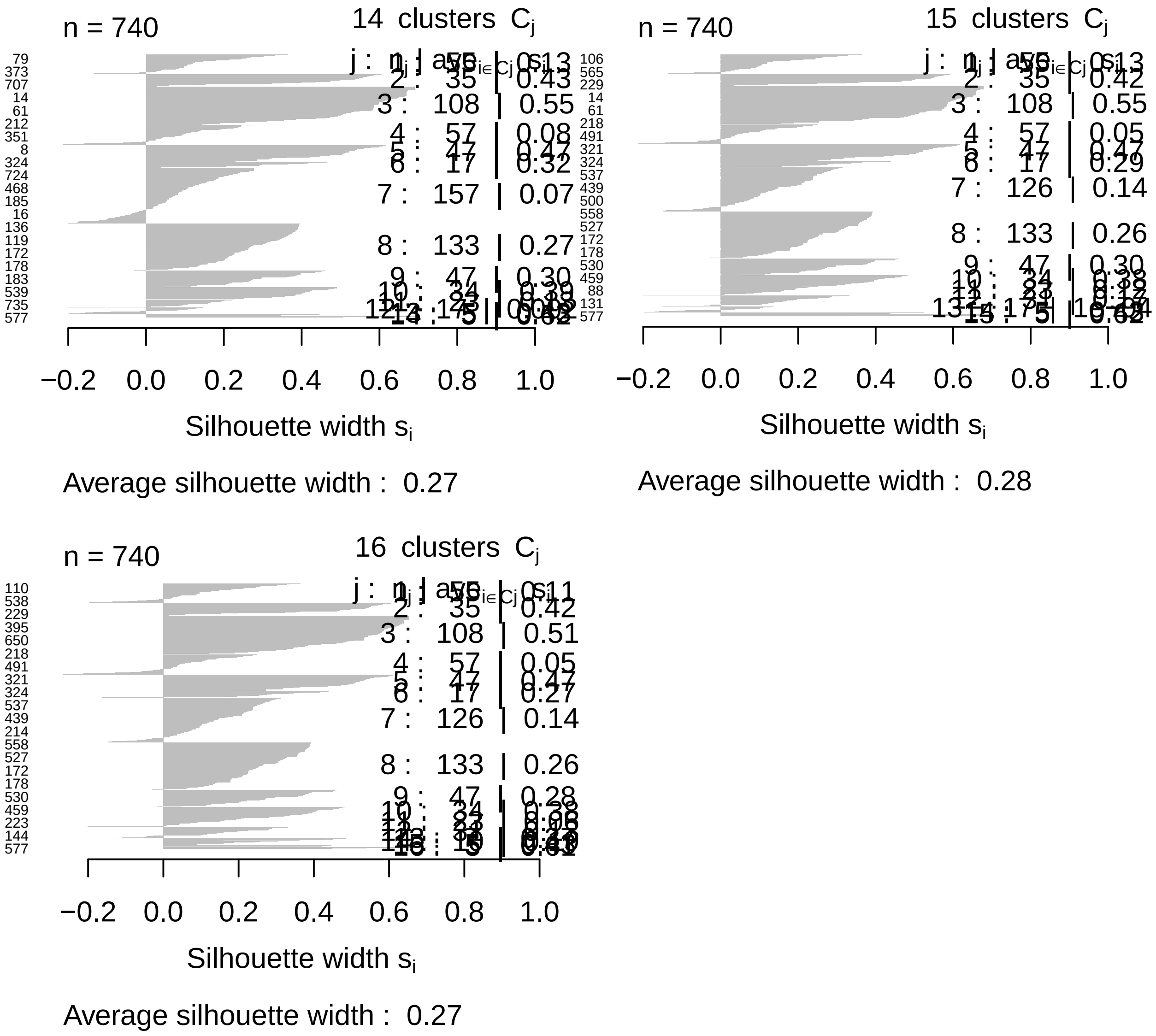

Figure 6: Silhouette plots for hierarchical clustering after PCA. k = 15 returned the maximum silhouette width values while the k = 14 and 16 are cluster sizes with similar silhouette widths.

Code

```
# ----- SETUP ----- #

packages <- c("knitr", "tidyverse", "ggplot2", "cluster", "fossil")

lapply(packages, library, character.only = TRUE)

# Read data, extract labels, and keep only quantitative data

absentData_full <- read.csv("Absenteeism_at_work.csv", sep = ";")
```

```
absentData_lab <- absentData_full$`Reason.for.absence`
absentData_notclean <- absentData_full %>%
  select(-c("Reason.for.absence","ID","Month.of.absence","Day.of.the.week","Seasons",
            "Disciplinary.failure","Education","Social.drinker","Social.smoker"))
# ----- DATA EXPLORATION ----- #
# Check data types, min, max, and missing data
data_type <- sapply(absentData_notclean,class)
min <- sapply(absentData_notclean, function(col){min(col,na.rm=TRUE)})
max <- sapply(absentData_notclean, function(col){max(col,na.rm=TRUE)})
nulls <- sapply(absentData_notclean, function(col){sum(is.na(col))})
blanks <- sapply(absentData_notclean,
                 function(col){ifelse(is.na(sum(col == "")), 0, sum(col == ""))})
data.frame(row.names = names(nulls), data_type=data_type, min=min, max=max,
           nulls_blanks=nulls+blanks)
# Create box plots to check for outliers
b01 <- boxplot(absentData_notclean$Transportation.expense, ylab = "Transportation expense")
b02 <- boxplot(absentData_notclean$Distance.from.Residence.to.Work,
               ylab = "Distance from residence to work")
b03 <- boxplot(absentData_notclean$Service.time, ylab = "Service time")
b04 <- boxplot(absentData_notclean$Age, ylab = "Age")
b05 <- boxplot(absentData_notclean$Work.load.Average.day,
               ylab = "Work load average per day")
b06 <- boxplot(absentData_notclean$Hit.target, ylab = "Hit target")
b07 <- boxplot(absentData_notclean$Son, ylab = "Son")
b08 <- boxplot(absentData_notclean$Pet, ylab = "Pet")
b09 <- boxplot(absentData_notclean$Weight, ylab = "Weight")
b10 <- boxplot(absentData_notclean$Height, ylab = "Height")
b11 <- boxplot(absentData_notclean$Body.mass.index, ylab = "Body mass index")
b12 <- boxplot(absentData_notclean$Absenteeism.time.in.hours,
               ylab = "Absenteeism time in hours")
# ----- DATA CLEANSING -----
# Handle outliers by capping them using interquartile range
cap <- function(val, bplot) {
  lower_fence <- bplot$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$stats[2])) #Q3+1.5*IQR
  val <- ifelse(val < lower_fence, lower_fence, val)
  val <- ifelse(val > upper_fence, upper_fence, val)
  val
}
absentData <- absentData_notclean %>%
  mutate(Transportation.expense = cap(val=Transportation.expense, bplot=b01),
         Service.time = cap(val=Service.time, bplot=b03),
         Age = cap(val=Age, bplot=b04),
         Work.load.Average.day = cap(val=Work.load.Average.day, bplot=b05),
         Hit.target = cap(val=Hit.target, bplot=b06),
         Pet = cap(val=Pet, bplot=b08),
         Height = cap(val=Height, bplot=b10),
         Absenteeism.time.in.hours = cap(val=Absenteeism.time.in.hours, bplot=b12))
# Scale the data
absentData_sd <- scale(absentData)
# ----- CLUSTERING FUNCTIONS ----- #
# Get silhouette for k-means clustering
kmcSilK <- function(k, data){
  x_k <- kmeans(data, k, nstart = 20)
  silhouette(x_k$cluster, dist(data))
}
# Get silhouette for hierarchical clustering
hcSilK <- function(k, data, method = "complete"){
  hc_out <- hclust(dist(data), method = method)
  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() +
    labs(x="Number of clusters (k)", y="Average silhouette width") +
    geom_point(data=max_point, colour="red") +
    geom_text(data=max_point, aes(label=ifelse(k==sil_max_k,max_lab,""),color="red"),
              hjust=1,vjust=1) +
    theme(legend.position="none")
  # Return plot, silhouettes, and max k
  list(plot=plot, sil_k=sil_k, max_k=sil_max_k)
}
# ----- AGGLOMERATIVE HIERARCHICAL CLUSTERING ----- #
# Get silhouette scores for multiple k values and plot them
set.seed(3)
```

```
k <- c(2:20)
hc_good_of_cluster <- chooseK(k, hcSilK, absentData_sd) # uses complete linkage
hc_good_of_cluster$plot
plotSil(hc_good_of_cluster$sil_k[[9]])
plotSil(hc_good_of_cluster$sil_k[[14]])
plotSil(hc_good_of_cluster$sil_k[[19]])
k <- 10
# Perform hierarchical clustering using best k
set.seed(3)
hc_out <- hclust(dist(absentData_sd))
ri_hc <- rand.index(cutree(hc_out, k), as.numeric(as.factor(absentData_lab)))
ari_hc <- 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)
kmc_good_of_cluster <- chooseK(k, kmcSilK, absentData_sd)
kmc_good_of_cluster$plot
# Based off silhouette plots, choose the best k
plotSil(kmc_good_of_cluster$sil_k[[15]])
plotSil(kmc_good_of_cluster$sil_k[[16]])
plotSil(kmc_good_of_cluster$sil_k[[17]])
k <- kmc_good_of_cluster$max_k
# Perform k-means clustering with best k and compute the rand indices
set.seed(3)
km_out <- kmeans(absentData, k, nstart = 20)
km_clusters <- km_out$cluster
ri_kmc <- rand.index(km_clusters, as.numeric(as.factor(absentData_lab)))
ari_kmc <- adj.rand.index(km_clusters, as.numeric(as.factor(absentData_lab)))
# ----- HIERARCHICAL CLUSTERING AFTER PCA ----- #
# Proportion of variance explained
```

```r
set.seed(3)
pr_out <- prcomp(absentData, scale = TRUE)
# Scree plot
pve <- 100 * pr_out$sdev^2 / sum(pr_out$sdev^2)
plot(pve, type = "o", cex.lab=0.75,
  xlab = "Principal Component", col = "blue",
  ylab = "Percent of variance explained (PVE)")
# Get silhouette scores for multiple k values and plot them
set.seed(3)
k <- c(2:20)
pcahc_good_of_cluster <- chooseK(k, hcSilK, pr_out$x[, 1:8])
pcahc_good_of_cluster$plot
# Based off silhouette plots, choose the best k
plotSil(pcahc_good_of_cluster$sil_k[[13]])
plotSil(pcahc_good_of_cluster$sil_k[[14]])
plotSil(pcahc_good_of_cluster$sil_k[[15]])
k <- pcahc_good_of_cluster$max_k
set.seed(3)
hc_out <- hclust(dist(dist(pr_out$x[, 1:8])))
hc_clusters <- cutree(hc_out, k)
ri_pcahc <- rand.index(hc_clusters, as.numeric(as.factor(absentData_lab)))
ari_pcahc <- adj.rand.index(hc_clusters, as.numeric(as.factor(absentData_lab)))
# ----- COMPARISON ----- #
clustering_method <- c("hierarchical", "k-means", "pca followed by hierarchical")
rand_index <- round(c(ri_hc, ri_kmc, ri_pcahc),3)
adj_rand_index <- round(c(ari_hc, ari_kmc, ari_pcahc),3)
comparison <- data.frame(clustering_method = clustering_method,
                         rand_index=rand_index, adj_rand_index=adj_rand_index)
kable(comparison)
```

References

Araujo, V. S., Rezende, T. S., Guimarães, A. J., Araujo, V. J. S., & Campos Souza, P. V. de. (2019). A hybrid approach of intelligent systems to help predict absenteeism at work in companies. In *SN applied sciences*. Springer. https://doi.org/10.1007/s42452-019-0536-y

Martiniano, A., & Ferreira, R. (2018). *Absenteeism at work*. UC Irvine Machine Learning Repository. https://doi.org/10.24432/C5X882

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