Compulsory exercise 1: Group 16

TMA4268 Statistical Learning V2022

Weicheng Hua, Emil Johannesen Haugstvedt, Torbjørn Baadsvik

29 March, 2022

Contents

Problem 1

a)

b)

c)

d)

Problem 2

a)

b)

Problem 3

a)

b)

Problem 4

a)

b)

c)

```
library(tidyverse)
library(palmerpenguins) # Contains the data set "penguins".
data(penguins)
names(penguins) <- c("species","island","billL","billD","flipperL","mass","sex","year")
Penguins_reduced <- penguins %>%
  dplyr::mutate(mass = as.numeric(mass),
         flipperL = as.numeric(flipperL),
         year = as.numeric(year)) %>%
  drop_na()
# We do not want "year" in the data (this will not help for future predictions)
Penguins_reduced <- Penguins_reduced[,-c(8)]
set.seed(4268)
# 70% of the sample size for training set
training_set_size <- floor(0.7 * nrow(Penguins_reduced))
train_ind <- sample(seq_len(nrow(Penguins_reduced)), size = training_set_size)
train <- Penguins_reduced[train_ind, ]
test <- Penguins_reduced[-train_ind, ]
```

d)

Problem 5

a)

i)	ii)	iii)	iv)
FALSE	TRUE	FALSE	TRUE

b)

```
svc.cvtune <- function(kernel, paramgrid, k){
 ctrl <- tune.control(sampling="cross", cross=k, nrepeat = 1)
 cvtune.result <- tune(method=svm, species~., kernel=eval(kernel),
      data=train, ranges=paramgrid, tunecontrol=ctrl)
 cvtune.result
}
reps <- 10
k <- 5
svc.fit_and_eval <- function(kernel, paramgrid, k){
  res <- svc.cvtune(kernel, paramgrid, k)
  print(strrep("_", 60))
  print(paste(eval(kernel), "support vector classifier"))
  print(strrep("-", 60))
  print("Parameters:")
  print(c(res$best.parameters))

  model <- res$best.model
  pred <- as.factor(predict(model, test[-c(1)]))
  cm <- confusionMatrix(data=pred, reference=test$species)
  print(cm)
}
#svc.fit_and_eval("linear", data.frame(cost=.1*1:20), k)
#svc.fit_and_eval("radial", data.frame(cost=.1*1:20, gamma=.1*1:20), k)
```

Using a linear rather than radial kernel in the support vector classifier yields slightly superior results on the test set. Thus, the linear kernel is preferred.

Problem 6

a)

```
id <- "1NJ1SuUBebl5P8rMSIwm_n3S8a7K43yP4" # google file ID
happiness <- read.csv(sprintf("https://docs.google.com/uc?id=%s&export=download", id),fileEncoding="UTF
colnames(happiness)
```

```
##  [1] "Country.name"
##  [2] "Regional.indicator"
```

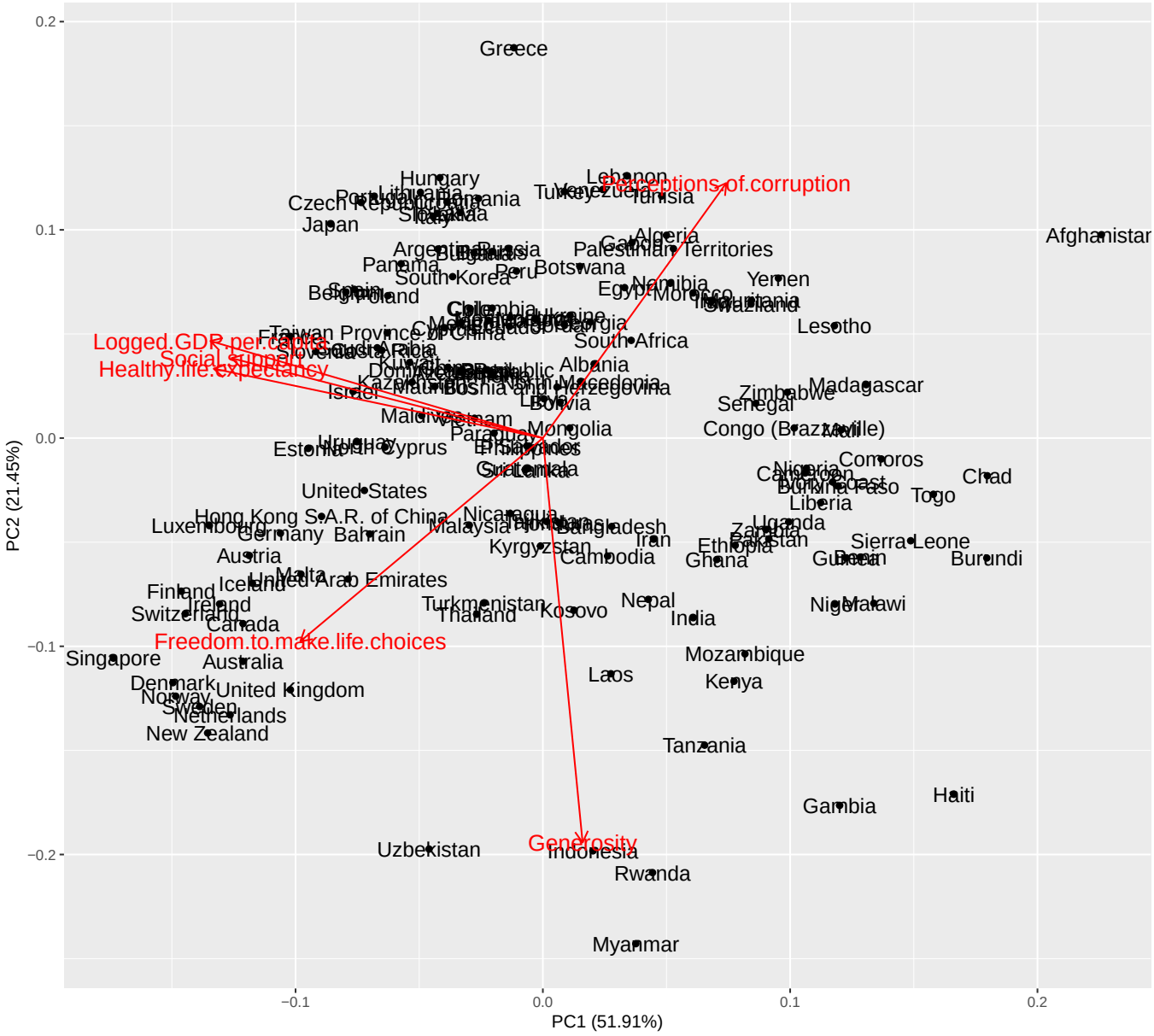
```
##  [3] "Ladder.score"
##  [4] "Standard.error.of.ladder.score"
##  [5] "upperwhisker"
##  [6] "lowerwhisker"
##  [7] "Logged.GDP.per.capita"
##  [8] "Social.support"
##  [9] "Healthy.life.expectancy"
## [10] "Freedom.to.make.life.choices"
## [11] "Generosity"
## [12] "Perceptions.of.corruption"
## [13] "Ladder.score.in.Dystopia"
## [14] "Explained.by..Log.GDP.per.capita"
## [15] "Explained.by..Social.support"
## [16] "Explained.by..Healthy.life.expectancy"
## [17] "Explained.by..Freedom.to.make.life.choices"
## [18] "Explained.by..Generosity"
## [19] "Explained.by..Perceptions.of.corruption"
## [20] "Dystopia...residual"
```

```
cols = c('Country.name',
         'Ladder.score',  # happiness score
         'Logged.GDP.per.capita',
         'Social.support',
         'Healthy.life.expectancy',
         'Freedom.to.make.life.choices',
         'Generosity',  # how generous people are
         'Perceptions.of.corruption')
# We continue with a subset of 8 columns:
happiness = subset(happiness, select = cols)
rownames(happiness) <- happiness[, c(1)]
# And we creat an X and a Y matrix
happiness.X = happiness[, -c(1, 2)]
happiness.Y = happiness[, c(1, 2)]
happiness.XY = happiness[, -c(1)]
# scale
happiness.X = data.frame(scale(happiness.X))
str(happiness)
```

```
## 'data.frame':    149 obs. of  8 variables:
##  $ Country.name                : chr  "Finland" "Denmark" "Switzerland" "Iceland" ...
##  $ Ladder.score                : num  7.84 7.62 7.57 7.55 7.46 ...
##  $ Logged.GDP.per.capita       : num  10.8 10.9 11.1 10.9 10.9 ...
##  $ Social.support              : num  0.954 0.954 0.942 0.983 0.942 0.954 0.934 0.908 0.948 0.934 ..
##  $ Healthy.life.expectancy     : num  72 72.7 74.4 73 72.4 73.3 72.7 72.6 73.4 73.3 ...
##  $ Freedom.to.make.life.choices: num  0.949 0.946 0.919 0.955 0.913 0.96 0.945 0.907 0.929 0.908 ...
##  $ Generosity                  : num  -0.098 0.03 0.025 0.16 0.175 0.093 0.086 -0.034 0.134 0.042 ..
##  $ Perceptions.of.corruption   : num  0.186 0.179 0.292 0.673 0.338 0.27 0.237 0.386 0.242 0.481 ...
```

```
library(ggfortify)
pca_mat = prcomp(happiness.X, center=T, scale=T)
# Score and loadings plot:
autoplot(pca_mat, data = happiness.X, colour='Black',
         loadings = TRUE, loadings.colour = 'red',
```

```
        loadings.label = TRUE, loadings.label.size = 5,
        label=T, label.size=4.5)
```



i)

We observe that the loading associated with the variable "Generosity" is approximately -0.2 for PC2, and 0.02 for PC1. The large difference in the absolute value of the two loadings imply that the "Generosity" variable has a greater impact on PC2 than PC1. Meanwhile, we observe that the loading associated with "Freedom.to.make.life.choices" is approximately -0.1 for both PC1 and PC2. Thus, this variable has nearly equal impact on PC1 and PC2.

ii)

Afghanistan appears to be clearly separated from the other countries in this plot, and may be considered to be an outlier.

b)

c)

d)