```r
library(tidyverse)
library(dplyr)
library(ggplot2)


############
#
# Task 5
#
############


# vector used for the plot of the density
x=seq(-5, 15, 0.1)

# sample size n=30, 100, 300
for(n in c(30, 100, 300)){

  # generate random numbers, remember: sd=sqrt(sigma^2)
  X = rnorm(n, mean=5, sd = 2)

  # plot the histogram
  hist(X, freq=FALSE)
  # add estimation and density to the previous plot
  lines(density(X), col="red")
  lines(x, dnorm(x, mean=5, sd=2), col = "blue")
  # alternatively you can use the function curve
}


############
#
# Task 6
#
############

library(mvtnorm)
library(MASS)

# (a)
I = matrix(c(1,0,0,0,
             0,1,0,0,
             0,0,1,0,
             0,0,0,1), nrow=4)
X = rmvnorm(100, mean=c(0,0,0,0), sigma=I)
# alternative
# X=matrix(rnorm(400), ncol=4)

# (b)
mu = c(1,0,2,-1)
Sigma1 = matrix(c(4,2,2,3,
                  2,3,2,1,
                  2,2,5,2,
                  3,1,2,3), nrow=4)
Sigma2 = matrix(c(4.5, 4.75, 2, 2.25,
                  4.75, 5.25, 2.75, 3.25,
                  2, 2.75, 2.75, 3.5,
                  2.25, 3.25, 3.5, 4.5), nrow=4)

# (c)
# singular-value decomposition (see Theorem I.1.5 of the lecture)
SVD1 = svd(Sigma1)
SVD2 = svd(Sigma2)
# Sigma2 has an eigenvalue equal to 0
SVD2
SVD2$d[4] = 0
# Computation of Sigma^(1/2)
Sigma1.half = SVD1$u%*%diag(sqrt(SVD1$d))%*%t(SVD1$v)
Sigma2.half = SVD2$u%*%diag(sqrt(SVD2$d))%*%t(SVD2$v)
# Transformation of multivariate normal distribution (see Corollary I.2.4 of the lecture)
Y = t(mu + Sigma1.half%*%t(X))
Z = t(mu + Sigma2.half%*%t(X))

# (d)
pairs(X)
pairs(Y)
pairs(Z)

#(e)
ginv(Sigma1)
solve(Sigma1)
ginv(Sigma2)
solve(Sigma2)


############
#
# Task 7
#
############

# (a)
# read in both data files
data.survey.a = read.csv2("Survey1a.csv",stringsAsFactors=TRUE)
data.survey.b = read.csv2("Survey1b.csv",stringsAsFactors=TRUE)
# Note: In some R versions, stringsAsFactors is by default TRUE, in others
# it is FALSE. Here we set it explicitly to ensure compatibility

# (b)
# selection of the columns
idx = (regexpr("Dim",names(data.survey.a))>-1)|names(data.survey.a)=="MeanScore"
# transformation from factor to character and from character to numeric
data.survey.a[, idx] <- sapply(sapply(data.survey.a[, idx], as.character), as.numeric)
data.survey.b[, idx] <- sapply(sapply(data.survey.b[, idx], as.character), as.numeric)

# (c)
# select indices of missing values in Survey1b.csv
missing.values = which(is.na(data.survey.b), arr.ind = TRUE)

for(row in missing.values[,1]){
  # assign data.survey.b with the corresponding values from data.survey.a
  # interviews can be identified via Person and Date uniquely
  data.survey.b[row,]=data.survey.a[data.survey.a$Person==data.survey.b$Person[row] & data.survey.a$Date == as.character(data.survey.b$Date[row]), ]
}
data.survey = rbind(data.survey.a, data.survey.b)
# delete duplicated rows
data.survey = data.survey[!duplicated(data.survey), ]


# alternative using the package tidyverse
data.survey = rbind(data.survey.a, data.survey.b)
data.survey = data.survey %>%
  # sort the the data by interviews
  arrange(Person, Date) %>%
  # duplicated interviews are in successive rows
  # fill missing values in data.survey.b with value above from data.survey.a
  fill(DimBody, DimEmotion, DimSelf, DimFamily, DimFriends, DimSchool, MeanScore)
# delete duplicated rows
data.survey = filter(data.survey,duplicated(data.survey)==FALSE)

# (d)
plot(data.survey$Age, data.survey$DimSchool, col=c("red","blue")[data.survey$Sex]);
legend(x="topright", legend = levels(data.survey$Sex), col=c("red","blue"), pch=1)

# alternative using the package tidyverse
ggplot(data.survey)+geom_point(aes(Age, DimSchool, col=Sex))

#(e)
boxplot(data.survey$DimFriends[data.survey$Sex=="m"],data.survey$DimFriends[data.survey$Sex=="f"], xaxt="n",main="Boxplot for dimension: Friends")
axis(1, at=1:2, labels=c('female','male'))

# alternative using the package tidyverse
ggplot(data.survey)+geom_boxplot(aes(DimFriends, col=Sex))

# (f)
save(data.survey,file="Survey1.RData")



############
#
# Task 8
#
############

# (a)
credits.data = read.csv("credits.wsv",sep=" ")

# (b)
# convert the coding of gastarb
credits.data$gastarb = credits.data$gastarb%%2+1

# (c)
# convert numeric to factor
dtime.as.factor = cut(credits.data$time,
                      breaks=c(0,6,12,18,24,30,36,42,48,54,Inf),
                      labels=(10:1))
credits.data$dtime = as.numeric(as.character(dtime.as.factor))

damount.as.factor = cut(credits.data$amount,
                        breaks=c(0,500,1000,1500,2500,5000,7500,10000,15000,20000,Inf),
                        labels=(10:1))
credits.data$damount = as.numeric(as.character(damount.as.factor))

dage.as.factor = cut(credits.data$age,
                        breaks=c(0,25,39,59,64,Inf),
                        labels=c(1,2,3,5,4))
credits.data$dage = as.numeric(as.character(dage.as.factor))

vars = names(credits.data)
# exclude attributes time, age, amount and repayment from the index vector
idx.vec = (1:ncol(credits.data))[!(vars=="time")+(vars=="age")+(vars=="amount")+(vars=="repayment")]

# sum of the remaining columns
score.vec = rowSums(credits.data[,idx.vec])
# add score to the data frame
credits.data$simple.score = score.vec

# (d)
mean(score.vec[credits.data$repayment==0])
mean(score.vec[credits.data$repayment==1])

# (e)
write.csv(credits.data,"credits.csv")
```