```
 #########################################
 #                                       #
###            DAD CA 1 2017            ###
 #                                       #
 #########################################

# The key to success in any organization is attracting and retaining top talent.
# You are an HR analyst at my company, and one of my tasks is to determine which
factors
# keep employees at my company and which prompt others to leave. We need to know what
# factors we can change to prevent the loss of good people.

# You have data about past and current employees in a spreadsheet. It has various data
# points on our employees, but we're' most interested in whether they're still with the
# company or whether they've gone to work somewhere else. And we want to understand how
# this relates to workforce attrition.

#Attributes:
 # Age: in years
 # Attrition: Y/N the dependent variable -- have they left the company?
 # BusinessTravel: Non-Travel; Traval_Frequently, Travel_Rarely
 # DailyRate: Consultancy Charge per Day
 # Department: Human Resources; Research & Development; Sales
 # DistanceFromHome: How far the employe lives from work
 # Education: 1 'Below College'; 2 'College'; 3 'Bachelor'; 4 'Master'; 5 'Doctor'
 # EducationField: Human Resources; Life Sciences; Marketing; Medical; Other; Technical
Degree
 # EmployeeCount: No of employes in this record
 # EmployeeNumber: Employee ID
 # EnvironmentSatisfaction: 4 point Likert scale: 1 'Low'; 2 'Medium'; 3 'High'; 4
'Very High'
 # Gender: Male / Female
 # HourlyRate: Consultancy Charge per Hour
 # JobInvolvement: 4 point Likert scale: 1 'Low'; 2 'Medium'; 3 'High'; 4 'Very High'
 # JobLevel     Metadata not available -- make an assumption
 # JobRole: Healthcare Representative;  Human Resources; Laboratory Technician;
Manager; Manufacturing Director; Research Director; Research Scientist; Sales
Executive; Sales Representative
 # JobSatisfaction: 4 point Likert scale: 1 'Low'; 2 'Medium'; 3 'High'; 4 'Very High'
 # MaritalStatus: Divorced; Married; Single
 # MonthlyIncome: monthly salary
 # MonthlyRate: Consultancy Charge per Day
 # NumCompaniesWorked: No. of previous employeers
 # Over18: Y/N
 # OverTime: Yes/No
 # PercentSalaryHike: Last Years Increment
 # PerformanceRating:  4 point Likert scale: 1 'Low'; 2 'Good'; 3 'Excellent'; 4
'Outstanding'
 # RelationshipSatisfaction:  4 point Likert scale: 1 'Low'; 2 'Medium'; 3 'High'; 4
'Very High'
 # StandardHours: Contract hours
 # StockOptionLevel: No available metadata -- make an assumption :)
 # TotalWorkingYears: Career Age
 # TrainingTimesLastYear: No. of training courses attended last year
 # WorkLifeBalance: 4 Point Likert Scale: 1 'Bad'; 2 'Good'; 3 'Better'; 4 'Best'
 # YearsAtCompany: Time spent with company
 # YearsInCurrentRole: Time in current role
 # YearsSinceLastPromotion: No. of years since last promoted
 # YearsWithCurrManager: Year spent with current manager

 setwd("/.../DAD/CA1") #change this to where you downloaded the .csv
 hrdata <- read.csv("CA1 Data.csv", stringsAsFactors = T) #will autoencode the text
attributes to factors

 #ok, now we need to make a dataset unique to you
 set.seed(1337) # <-- put your student number here WITHOUT the x!! Leave off a starting
zero if you have one
 #e.g.: set.seed(62345678)
```

```r
index <- sample(1:nrow(hrdata), 400, replace=FALSE)
my_ca_dataset <- hrdata[index, ] # here we're subsetting your part of the dataset

#Unfortunately, due to a technical error, 15 columns of the data were lost :(
#HR blamed IT, IT blamed HR, your manager will blame you, so let's just hope those
columns weren't important!
index2 <- sample(1:(ncol(my_ca_dataset)-3), 15, replace=FALSE)
index2 <- index2 + 3 #the minus and then plus 3 protects the first 3 columns of the
dataset

print(paste("I lost:", names(my_ca_dataset)[index2]))

my_ca_dataset <- my_ca_dataset[, -index2]

# Unfortunately, there was another incident. The intern split their coffee
# on your keyboard and may have deleted data from a number of the remaining columns

v <- round(runif(ncol(my_ca_dataset), min=1, max=6))
v <- cut(v, breaks = c(0,4,max(v)), labels = c("a","b"))
v[2] <- "a"

Pna <- runif(1000, min=0, max=0.18)
Pna <- Pna - .03
Pna[Pna < 0] <- 0

for (i in 1:length(v)) {
  if (v[i] == "b") {
    nadex <- sample(1:nrow(my_ca_dataset), nrow(my_ca_dataset) *
Pna[sample(1:length(Pna), 1, replace=FALSE)], replace=FALSE)
    my_ca_dataset[nadex, i] <- NA
    v[i] <- "a"
  }
}

# Clean up

rm(v)
rm(Pna)
rm(hrdata)
rm(index)
rm(index2)
rm(i)
rm(nadex)

###### Backup your data set

#In case anything goes wrong, we'll store a copy in memory
my_ca_backup <- my_ca_dataset

write.csv(file="my_ca_dataset.csv", my_ca_dataset, row.names = F)
# If you mess up uncomment and run to restore the original dataset:
# my_ca_dataset <- my_ca_backup

#Now please begin, and good luck!

##########################################
# Begin Foundations
str(my_ca_dataset)

# F1 ####################################

#my_ca_dataset$Education <- factor(my_ca_dataset$Education, levels = c(1,2,3,4,5),
labels=c("BC", "C", "UG", "MSc", "PhD"))
my_ca_dataset$EnvironmentSatisfaction <- factor(my_ca_dataset$EnvironmentSatisfaction,
levels = c(1:4), labels=c("Low", "Medium", "High", "v. High"))
my_ca_dataset$JobInvolvement <- factor(my_ca_dataset$JobInvolvement, levels = c(1:4),
labels=c("Low", "Medium", "High", "v. High"))
```

```
 #my_ca_dataset$JobLevel <- factor(my_ca_dataset$JobLevel) #insufficient information to
do more
 #my_ca_dataset$JobSatisfaction <- factor(my_ca_dataset$JobSatisfaction, levels =
c(1:4), labels=c("Low", "Medium", "High", "v. High"))
 my_ca_dataset$PerformanceRating <- factor(my_ca_dataset$PerformanceRating, levels =
c(1:4), labels=c("Low", "Good", "Excellent", "Outstanding"))
 my_ca_dataset$RelationshipSatisfaction <-
factor(my_ca_dataset$RelationshipSatisfaction, levels = c(1:4), labels=c("Low",
"Medium", "High", "v. High"))
 my_ca_dataset$StockOptionLevel <- factor(my_ca_dataset$StockOptionLevel) #don't have
more information
 #my_dataset$WorkLifeBalance <- factor(my_ca_dataset$WorkLifeBalance, levels = c(1:4),
labels=c("Bad", "Good", "Better", "Best"))

 # I'm also going to get rid of 3 pointless columns:
 my_ca_dataset <- my_ca_dataset[ , -(c(5,6,13,16))]
 # Employee number doesn't really tell us much for any question in the CA
 # Employee count is alwats 1
 # Over18 is always yes
 # standard hours is always 80

 # F2 #####################################

 #strictly speaking this would tell us
 summary(my_ca_dataset)

 #alternatively, we also do:
 sapply(my_ca_dataset,function(x) sum(is.na(x)))

 # F3 #####################################

 # Basic Option: delete a column (if you deleted rows, that's fine too)
 my_ca_dataset <- my_ca_dataset[, -16] # YearsWithCurrManager
 sapply(my_ca_dataset,function(x) sum(is.na(x)))
 #YearsWithCurrManager is gone

 # Intemediate option
 my_ca_dataset$DailyRate[is.na(my_ca_dataset$DailyRate)] <-
mean(my_ca_dataset$DailyRate, na.rm = T)
 sapply(my_ca_dataset,function(x) sum(is.na(x)))

 # Advanced option: use mice
 library(mice)
 mice_mod <- mice(my_ca_dataset[, !names(my_ca_dataset) %in%

c('BusinessTravel','JobInvolvement','MaritalStatus','MonthlyIncome','NumCompaniesWorked',
 'PerformanceRating', 'StockOptionLevel')], method='rf')

 mice_output <- complete(mice_mod)
 my_ca_dataset$Age <- mice_output$Age
 sapply(my_ca_dataset,function(x) sum(is.na(x)))

 # F4 #####################################

 #There are a few different ways of doing this. The question asks us to only locate if
there may be missing values.
 #First things first, a categorical cannot by definition have outlier values.

 numericCols <- c(1, 4, 8:10, 14, 15)
 boxplot(my_ca_dataset[, numericCols])

 #something like this was fine for 1 point

 # so this sort of gives us an intutition that there may be some outliers (values
beyond the whisker),
 # but because of the variation in range of the data, it's hard to really see what's
going on.
```

```
AgeOutliers <- boxplot.stats(my_ca_dataset$Age)$out
AgeOutliers #no outliers

#this would show how to identify outliers in general, this would have been fine for 2
points

name <- c()
amount <- c()

vNames <- names(my_ca_dataset[, numericCols])

for (i in 1:length(vNames)) {
  outliers <- boxplot.stats(my_ca_dataset[, vNames[i]])$out
  if (length(outliers) > 0) {
    name <- c(name, vNames[i])
    amount <- c(amount, length(outliers))
  }
}

df <- data.frame(name, amount)
df

#i would have been fine with you also copy/pasting the answer for 2 points and
changing the column names

# End Foundations
##########################################

# Make a backup
# to store a copy of the data.frame to disk:
save(my_ca_dataset, file="my_ca_dataset.RData")

# if at any point, you need to restore your dataset run:
# load("my_ca_dataset.RData")

##########################################

# Question: B1
summary(my_ca_dataset$Age) #min, max, mean, and median
sd(my_ca_dataset$Age) #standard deviation

##########################################

# Question: B2
table(my_ca_dataset$Attrition)

##########################################

# Question: B3
#according to B2, more people have remained than left.
#For completeness
barplot(table(my_ca_dataset$Attrition))

##########################################

# Question: B4
barplot(table(my_ca_dataset$BusinessTravel))
#most employees travel rarely, some travel frequently, and very few have no travel

##########################################

# Question: I1a
# Do people who leave the company, have on average higher rates of incomes? Briefly
note if the answer is what you expected.
aggregate(MonthlyIncome ~ Attrition, mean, data = my_ca_dataset)
#nope those that leave seem to earn less on average -- probably not that surprising

##########################################
```

```
# Question: I1b
#if i'm concerned about time, i can actually more or less reuse my previous answer
here
aggregate(Age ~ Attrition, mean, data = my_ca_dataset)
#are those that leave are on average younger? Yes

##########################################

# Question: I2a
boxplot(Age ~ EnvironmentSatisfaction, data = my_ca_dataset)
#on average it seems that there is a tendancy for older members of staff to be happy
with the environment

##########################################

# Question: I2b
plot(my_ca_dataset$Age, my_ca_dataset$MonthlyIncome)
#this one is hard. It seems that younger members of staff tend to earn less.
#Older members of staff have a much greater range of earnings, but there doesn't
appear to be an obvious relationship
#at least without further analysis

##########################################

# Question: I3
#different ways to answer this. Easiest way to get an idea is
boxplot(my_ca_dataset$Age ~ my_ca_dataset$Attrition)
#we see that on average younger staff leave, so there probably is some relationship

##########################################

# Question: A1a (reusing the plot at the end of lab 2)
coplot(MonthlyIncome ~ BusinessTravel | Attrition, data = my_ca_dataset, panel =
panel.smooth, rows = 1)
#In general those that leave have a lower monthly income
#However, there doesn't seem to be much of a relationship between the amount or
travel.

#reusing a plot from lab 3
library(ggplot2)
library(ggthemes)
ggplot(my_ca_dataset, aes(x = NumCompaniesWorked, fill = Attrition)) +
  geom_bar(stat='count', position='dodge') +
  labs(x = 'NumCompaniesWorked') +
  facet_grid(.~JobInvolvement) +
  theme_few()

#No. of companies doesn't appear to make much difference
#However, it seems employees that have high job involvement are more likely to leave

##########################################

# Question:
#Note, because I have done all options for F3 I have no columns with missing values.
#If i still had missing values I'd need to remove them for this to work

library(randomForest)
rf <- randomForest(Attrition ~., data = my_ca_dataset)
varImpPlot(rf)

#Most important seem to be: Monthly income, Total working years, Age, monthly rate
(which is essentially the same as monthly income)

##########################################

# Question: A3
summary(my_ca_dataset$Age)
```

```
 my_ca_dataset$NewAge <- cut(my_ca_dataset$Age, breaks = c(0, 25, 35, 61), labels =
c("Young", "Medium", "Older"))
 table(my_ca_dataset$NewAge, my_ca_dataset$BusinessTravel)
 #this would have been fine for a correct answer, but
 #a table  is somewhat misleading
 #we have more older employees than younger and medium employees.
 #A crosstable is more meaningful

 library(gmodels)
 CrossTable(my_ca_dataset$NewAge, my_ca_dataset$BusinessTravel, prop.chisq = F, prop.c
= F)

 #the row totals are important
 #77% of young travel rarely
 #73% of medium also travel rarely
 #70% of older travel rarely
 #where the rarely travel percentage goes down with age, the travel frequently goes up.
So older tend to travel more.

 ##########################################

 # Question: SO1
 library(caret)
 sample <- createDataPartition(my_ca_dataset$Attrition, p = .75, list = FALSE)
 train <- my_ca_dataset[sample, ]
 test <- my_ca_dataset[-sample, ]

 logit <- glm(train$Attrition ~.,family=binomial(link='logit'),data=train)

 results.1.logit <- predict(logit,newdata=test[,-2],type='response')
 #returns a probability that someone as left
 results.1.logit <- ifelse(results.1.logit > 0.5,"Yes","No")
 #if the probability is greater than .5 (50%) encode it as yes, otherwise no.
 #we can make the model more conservative by increasing this threshold
 (logitAcc1 <- 1- mean(results.1.logit != test$Attrition))
 #compute an accuracy for that

 #So accuracy is 88.9% is that good? Well:
 prop.table(table(test$Attrition))

 #Well, it's ok, had we always said no, we would also have got 86%.

 ##########################################

 # Question: SO2 -- let's reuse A2
 rf <- randomForest(Attrition ~., data = train, ntree=500, importance=T)
 rf_prediction <- predict(rf, newdata = test[, -2])
 (rfAccuracy <- 1- mean(rf_prediction != test$Attrition))

 #Performance isn't great, it's pretty similar to always saying no

 ##########################################

 # Question: SO3
 library(class)
 normalize <- function(x) { return ((x - min(x)) / (max(x) - min(x))) }
 #you'll see this next semester. But clustering requires data to be normalised (in the
same value range)
 #the normalise function above z-scores the data
 my_ca_dataset_n <- my_ca_dataset[, numericCols]
 summary(my_ca_dataset_n)
 my_ca_dataset_n <- as.data.frame(lapply(my_ca_dataset_n, normalize))
 summary(my_ca_dataset_n)
 #now all variables have been transformed to be within the same value range, i.e.
between 0 and 1
 #this is important because clustering uses notions of distance so all points need to
be within the same value range
```

```
library(clusterSim)
kmeansScores <- c()
for (i in 2:10) {
  clusters <- kmeans(my_ca_dataset_n, i)
  name <- paste0("kmeans", i)
  dbindex <- index.DB(my_ca_dataset_n, clusters$cluster, centrotypes="centroids")
  kmeansScores <- rbind(kmeansScores, dbindex$DB)
}

row.names(kmeansScores) <- c(2:10)
colnames(kmeansScores) <- c("k")

plot(kmeansScores, xlab="k", ylab="DBIndex")
library(fpc)
plotcluster(my_ca_dataset_n, kmeansClusters[["kmeans4"]]$cluster)

#the plot cluster essentially shows how the cluster labels do/don't overlap across the
dataset
#be careful interpreting this!! dc1 and dc2 are mutations of the data -- each one
actually represents
#more than one attribute of the data.
```