```
# Identifying Customer Targets (R)

# call in R packages for use in this study
library(lattice)  # multivariate data visualization
library(vcd)  # data visualization for categorical variables
library(ROCR)  # evaluation of binary classifiers

# read bank data into R, creating data frame bank
# note that this is a semicolon-delimited file
bank <- read.csv("/Users/neha/Documents/Github/DS680/MDS_chapter_3/MDS_Chapter_3/bank.csv",
sep = ";", stringsAsFactors = FALSE)

# This is  the structure of the bank data frame
print(str(bank))

# look at the first few rows of the bank data frame
print(head(bank))

# look at the list of column names for the variables
print(names(bank))

# look at class and attributes of one of the variables
print(class(bank$age))
print(attributes(bank$age))  # NULL means no special attributes defined
# plot a histogram for this variable
with(bank, hist(age))

# examine the frequency tables for categorical/factor variables
# showing the number of observations with missing data (if any)

print(table(bank$job , useNA = c("always")))
print(table(bank$marital , useNA = c("always")))
print(table(bank$education , useNA = c("always")))
print(table(bank$default , useNA = c("always")))
print(table(bank$housing , useNA = c("always")))
print(table(bank$loan , useNA = c("always")))

# Type of job (admin., unknown, unemployed, management,
# housemaid, entrepreneur, student, blue-collar, self-employed,
# retired, technician, services)
# put job into three major categories defining the factor variable jobtype
# the "unknown" category is how missing data were coded for job...
# include these in "Other/Unknown" category/level
white_collar_list <- c("admin.","entrepreneur","management","self-employed")
blue_collar_list <- c("blue-collar","services","technician")
bank$jobtype <- rep(3, length = nrow(bank))
bank$jobtype <- ifelse((bank$job %in% white_collar_list), 1, bank$jobtype)
bank$jobtype <- ifelse((bank$job %in% blue_collar_list), 2, bank$jobtype)
bank$jobtype <- factor(bank$jobtype, levels = c(1, 2, 3),
    labels = c("White Collar", "Blue Collar", "Other/Unknown"))
with(bank, table(job, jobtype, useNA = c("always")))  # check definition

# define factor variables with labels for plotting
bank$marital <- factor(bank$marital,
    labels = c("Divorced", "Married", "Single"))
bank$education <- factor(bank$education,
    labels = c("Primary", "Secondary", "Tertiary", "Unknown"))
bank$default <- factor(bank$default, labels = c("No", "Yes"))
bank$housing <- factor(bank$housing, labels = c("No", "Yes"))
bank$loan <- factor(bank$loan, labels = c("No", "Yes"))
bank$response <- factor(bank$response, labels = c("No", "Yes"))
```

```
# select subset of cases never perviously contacted by sales
# keeping variables needed for modeling
bankdata <- subset(bank, subset = (previous == 0),
    select = c("response", "age", "jobtype", "marital", "education",
               "default", "balance", "housing", "loan"))

# examine the structure of the bank data frame
print(str(bankdata))

# look at the first few rows of the bank data frame
print(head(bankdata))

# compute summary statistics for initial variables in the bank data frame
print(summary(bankdata))

# -----------------
# age  Age in years
# -----------------
# examine relationship between age and response to promotion
pdf(file = "fig_targeting_customers_age_lattice.pdf",
    width = 8.5, height = 8.5)
lattice_plot_object <- histogram(~age | response, data = bankdata,
    type = "density", xlab = "Age of Bank Client", layout = c(1,2))
print(lattice_plot_object)  # responders tend to be older
dev.off()

# ------------------------------------------------------------
# education
# Level of education (unknown, secondary, primary, tertiary)
# ------------------------------------------------------------
# examine the frequency table for education
# the "unknown" category is how missing data were coded
with(bankdata, print(table(education, response, useNA = c("always"))))

# create a mosaic plot in using vcd package
pdf(file = "fig_targeting_customers_education_mosaic.pdf",
    width = 8.5, height = 8.5)
mosaic( ~ response + education, data = bankdata,
  labeling_args = list(set_varnames = c(response = "Response to Offer",
  education = "Education Level")),
  highlighting = "education",
  highlighting_fill = c("cornsilk","violet","purple","white",
      "cornsilk","violet","purple","white"),
  rot_labels = c(left = 0, top = 0),
  pos_labels = c("center","center"),
  offset_labels = c(0.0,0.6))
dev.off()

# ---------------------------------------------------------------
# job status using jobtype
# White Collar: admin., entrepreneur, management, self-employed
# Blue Collar: blue-collar, services, technician
# Other/Unknown
# ---------------------------------------------------------------
# review the frequency table for job types
with(bankdata, print(table(jobtype, response, useNA = c("always"))))
pdf(file = "fig_targeting_customers_jobtype_mosaic.pdf",
    width = 8.5, height = 8.5)
mosaic( ~ response + jobtype, data = bankdata,
  labeling_args = list(set_varnames = c(response = "Response to Offer",
  jobtype = "Type of Job")),
  highlighting = "jobtype",
  highlighting_fill = c("cornsilk","violet","purple",
```

```
      "cornsilk","violet","purple"),
  rot_labels = c(left = 0, top = 0),
  pos_labels = c("center","center"),
  offset_labels = c(0.0,0.6))
dev.off()

# ------------------------------------------
# marital status
# Marital status (married, divorced, single)
# [Note: ``divorced'' means divorced or widowed]
# ------------------------------------------
# examine the frequency table for marital status
# anyone not single or married was classified as "divorced"
with(bankdata, print(table(marital, response, useNA = c("always"))))
pdf(file = "fig_targeting_customers_marital_mosaic.pdf",
    width = 8.5, height = 8.5)
mosaic( ~ response + marital, data = bankdata,
  labeling_args = list(set_varnames = c(response = "Response to Offer",
  marital = "Marital Status")),
  highlighting = "marital",
  highlighting_fill = c("cornsilk","violet","purple",
      "cornsilk","violet","purple"),
  rot_labels = c(left = 0, top = 0),
  pos_labels = c("center","center"),
  offset_labels = c(0.0,0.6))
dev.off()

# ---------------------------------------
# default  Has credit in default? (yes, no)
# ---------------------------------------
with(bankdata, print(table(default, response, useNA = c("always"))))
pdf(file = "fig_targeting_customers_default_mosaic.pdf",
    width = 8.5, height = 8.5)
mosaic( ~ response + default, data = bankdata,
  labeling_args = list(set_varnames = c(response = "Response to Offer",
  default = "Has credit in default?")),
  highlighting = "default",
  highlighting_fill = c("cornsilk","violet"),
  rot_labels = c(left = 0, top = 0),
  pos_labels = c("center","center"),
  offset_labels = c(0.0,0.6))
dev.off()

# ---------------------------------------
# balance  Average yearly balance (in Euros)
# ---------------------------------------
# examine relationship between age and response to promotion
pdf(file = "fig_targeting_customers_balance_lattice.pdf",
    width = 8.5, height = 8.5)
lattice_plot_object <- histogram(~balance | response, data = bankdata,
    type = "density",
    xlab = "Bank Client Average Yearly Balance (in dollars)",
    layout = c(1,2))
print(lattice_plot_object)  # responders tend to be older
dev.off()

# ------------------------------------
# housing  Has housing loan? (yes, no)
# ------------------------------------
with(bankdata, print(table(housing, response, useNA = c("always"))))
pdf(file = "fig_targeting_customers_housing_mosaic.pdf",
    width = 8.5, height = 8.5)
mosaic( ~ response + housing, data = bankdata,
```

```
  labeling_args = list(set_varnames = c(response = "Response to Offer",
  housing = "Has housing loan?")),
  highlighting = "housing",
  highlighting_fill = c("cornsilk","violet"),
  rot_labels = c(left = 0, top = 0),
  pos_labels = c("center","center"),
  offset_labels = c(0.0,0.6))
dev.off()

# ----------------------------------
# loan  Has personal loan? (yes, no)
# ----------------------------------
with(bankdata, print(table(loan, response, useNA = c("always"))))
pdf(file = "fig_targeting_customers_loan_mosaic.pdf",
    width = 8.5, height = 8.5)
mosaic( ~ response + loan, data = bankdata,
  labeling_args = list(set_varnames = c(response = "Response to Offer",
  loan = "Has personal loan?")),
  highlighting = "loan",
  highlighting_fill = c("cornsilk","violet"),
  rot_labels = c(left = 0, top = 0),
  pos_labels = c("center","center"),
  offset_labels = c(0.0,0.6))
dev.off()

# ----------------------------------
# specify predictive model
# ----------------------------------
bank_spec <- {response ~ age + jobtype + education + marital +
    default + balance + housing + loan}

# ----------------------------------
# fit logistic regression model
# ----------------------------------
bank_fit <- glm(bank_spec, family=binomial, data=bankdata)
print(summary(bank_fit))
print(anova(bank_fit, test="Chisq"))

# compute predicted probability of responding to the offer
bankdata$Predict_Prob_Response <- predict.glm(bank_fit, type = "response")

pdf(file = "fig_targeting_customer_log_reg_density_evaluation.pdf",
    width = 8.5, height = 8.5)
plotting_object <- densityplot( ~ Predict_Prob_Response | response,
               data = bankdata,
               layout = c(1,2), aspect=1, col = "darkblue",
               plot.points = "rug",
               strip=function(...) strip.default(..., style=1),
               xlab="Predicted Probability of Responding to Offer")
print(plotting_object)
dev.off()

# predicted response to offer using using 0.5 cut-off
# notice that this does not work due to low base rate
# we get more than 90 percent correct with no model
# (predicting all NO responses)
# the 0.50 cutoff yields all NO predictions
bankdata$Predict_Response <-
    ifelse((bankdata$Predict_Prob_Response > 0.5), 2, 1)
bankdata$Predict_Response <- factor(bankdata$Predict_Response,
    levels = c(1, 2), labels = c("NO", "YES"))
confusion_matrix <- table(bankdata$Predict_Response, bankdata$response)
cat("\nConfusion Matrix (rows=Predicted Response, columns=Actual Choice\n")
```

```
print(confusion_matrix)
predictive_accuracy <- (confusion_matrix[1,1] + confusion_matrix[2,2])/
                        sum(confusion_matrix)
cat("\nPercent Accuracy: ", round(predictive_accuracy * 100, digits = 1))

# this problem requires either a much lower cut-off
# or other criteria for evaluation... let's try 0.10 (10 percent cut-off)
bankdata$Predict_Response <-
    ifelse((bankdata$Predict_Prob_Response > 0.08), 2, 1)
bankdata$Predict_Response <- factor(bankdata$Predict_Response,
    levels = c(1, 2), labels = c("NO", "YES"))
confusion_matrix <- table(bankdata$Predict_Response, bankdata$response)
cat("\nConfusion Matrix (rows=Predicted Response, columns=Actual Choice\n")
print(confusion_matrix)
predictive_accuracy <- (confusion_matrix[1,1] + confusion_matrix[2,2])/
                        sum(confusion_matrix)
cat("\nPercent Accuracy: ", round(predictive_accuracy * 100, digits = 1))
# mosaic rendering of the classifier with 0.10 cutoff
with(bankdata, print(table(Predict_Response, response, useNA = c("always"))))
pdf(file = "fig_targeting_customers_confusion_mosaic_10_percent.pdf",
    width = 8.5, height = 8.5)
mosaic( ~ Predict_Response + response, data = bankdata,
  labeling_args = list(set_varnames =
  c(Predict_Response =
      "Predicted Response to Offer (10 percent cut-off)",
       response = "Actual Response to Offer")),
  highlighting = c("Predict_Response", "response"),
  highlighting_fill = c("green","cornsilk","cornsilk","green"),
  rot_labels = c(left = 0, top = 0),
  pos_labels = c("center","center"),
  offset_labels = c(0.0,0.6))
dev.off()

# compute lift using prediction() from ROCR and plot lift chart
bankdata_prediction <-
    prediction(bankdata$Predict_Prob_Response, bankdata$response)
bankdata_lift <- performance(bankdata_prediction , "lift", "rpp")
pdf(file = "fig_targeting_customers_lift_chart.pdf",
    width = 8.5, height = 8.5)
plot(bankdata_lift,
col = "blue", lty = "solid", main = "", lwd = 2,
    xlab = paste("Proportion of Clients Ordered by Probability",
    " to Subscribe\n(from highest to lowest)", sep = ""),
    ylab = "Lift over Baseline Subscription Rate")
dev.off()

# ------------------------------------------------------
# direct calculation of lift (code revised from textbook)
baseline_response_rate <-
    as.numeric(table(bankdata$response)[2])/nrow(bankdata)

lift <- function(x, baseline_response_rate) {
    mean(x) / baseline_response_rate
    }

decile_break_points <- c(as.numeric(quantile(bankdata$Predict_Prob_Response,
    probs=seq(0, 1, 0.10))))

bankdata$decile <- cut(bankdata$Predict_Prob_Response,
    breaks = decile_break_points,
    include.lowest=TRUE,
    labels=c("Decile_10","Decile_9","Decile_8","Decile_7","Decile_6",
    "Decile_5","Decile_4","Decile_3","Decile_2","Decile_1"))
```

```
# define response as 0/1 binary
bankdata$response_binary <- as.numeric(bankdata$response) - 1

cat("\nLift Chart Values by Decile:\n")
print(by(bankdata$response_binary, bankdata$decile,
    function(x) lift(x, baseline_response_rate)))

# Suggestions for the student:
# Try alternative methods of classification, such as neural networks,
# support vector machines, and random forests. Compare the performance
# of these methods against logistic regression. Use alternative methods
# of comparison, including area under the ROC curve.
# Ensure that the evaluation is carried out using a training-and-test
# regimen, perhaps utilizing multifold cross-validation.
# Check out the R package cvTools for doing this work.
# Examine the importance of individual explanatory variables
# in identifying targets. This may be done by looking at tests of
# statistical significance, classification trees, or random-forests-
# based importance assessment.
```