MDS_Exhibit_3_1.R

Neha

Wed Feb 15 16:07:16 2017

```
# 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
```

```
## Warning: package 'vcd' was built under R version 3.2.5
```

```
## Loading required package: grid
```

```
library(ROCR)  # evaluation of binary classifiers
```

```
## Loading required package: gplots
```

```
## Warning: package 'gplots' was built under R version 3.2.4
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
# 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 = ";",

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

```
## 'data.frame':    4521 obs. of  17 variables:
##  $ age      : int  30 33 35 30 59 35 36 39 41 43 ...
##  $ job      : chr  "unemployed" "services" "management" "management" ...
##  $ marital  : chr  "married" "married" "single" "married" ...
##  $ education: chr  "primary" "secondary" "tertiary" "tertiary" ...
##  $ default  : chr  "no" "no" "no" "no" ...
##  $ balance  : int  1787 4789 1350 1476 0 747 307 147 221 -88 ...
##  $ housing  : chr  "no" "yes" "yes" "yes" ...
##  $ loan     : chr  "no" "yes" "no" "yes" ...
##  $ contact  : chr  "cellular" "cellular" "cellular" "unknown" ...
##  $ day      : int  19 11 16 3 5 23 14 6 14 17 ...
##  $ month    : chr  "oct" "may" "apr" "jun" ...
##  $ duration : int  79 220 185 199 226 141 341 151 57 313 ...
##  $ campaign : int  1 1 1 4 1 2 1 2 2 1 ...
```

```
##  $ pdays    : int  -1 339 330 -1 -1 176 330 -1 -1 147 ...
##  $ previous : int  0 4 1 0 0 3 2 0 0 2 ...
##  $ poutcome : chr  "unknown" "failure" "failure" "unknown" ...
##  $ response : chr  "no" "no" "no" "no" ...
## NULL
```

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

```
##   age         job marital education default balance housing loan  contact
## 1  30  unemployed married   primary      no    1787      no   no cellular
## 2  33    services married secondary      no    4789     yes  yes cellular
## 3  35  management  single  tertiary      no    1350     yes   no cellular
## 4  30  management married  tertiary      no    1476     yes  yes  unknown
## 5  59 blue-collar married secondary      no       0     yes   no  unknown
## 6  35  management  single  tertiary      no     747      no   no cellular
##   day month duration campaign pdays previous poutcome response
## 1  19   oct       79        1    -1        0  unknown       no
## 2  11   may      220        1   339        4  failure       no
## 3  16   apr      185        1   330        1  failure       no
## 4   3   jun      199        4    -1        0  unknown       no
## 5   5   may      226        1    -1        0  unknown       no
## 6  23   feb      141        2   176        3  failure       no
```

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

```
##  [1] "age"       "job"       "marital"   "education" "default"
##  [6] "balance"   "housing"   "loan"      "contact"   "day"
## [11] "month"     "duration"  "campaign"  "pdays"     "previous"
## [16] "poutcome"  "response"
```

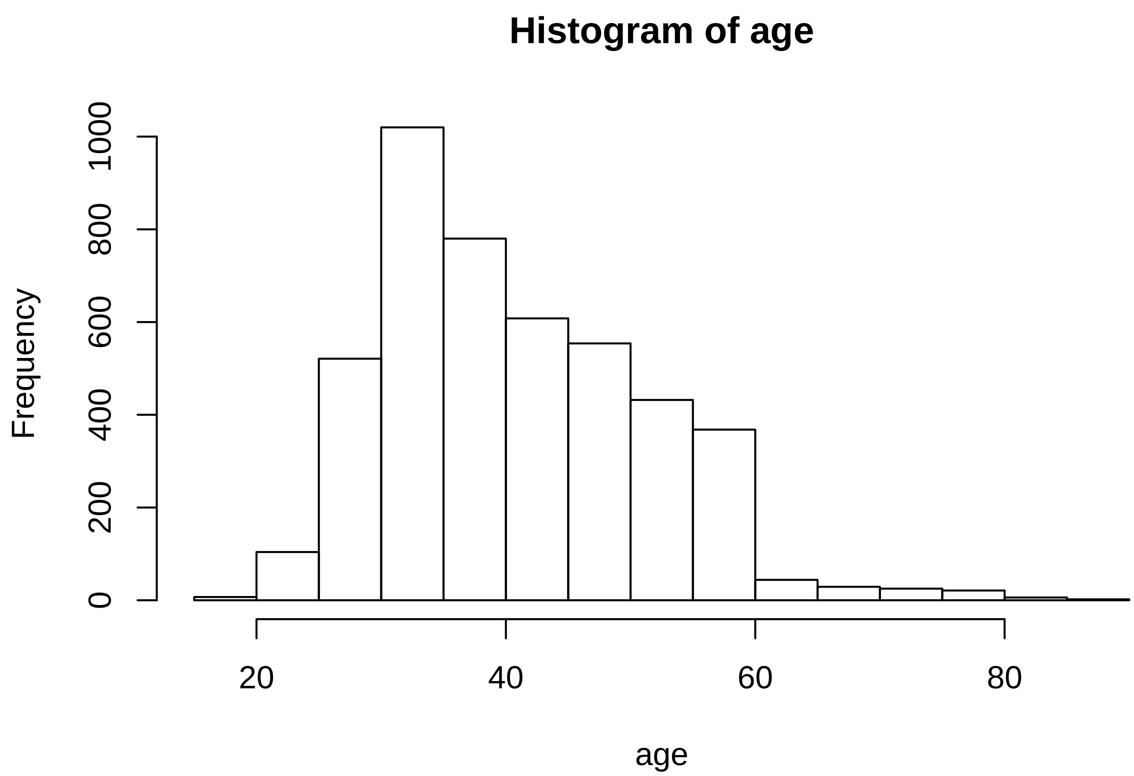
```r
# look at class and attributes of one of the variables
print(class(bank$age))
```

```
## [1] "integer"
```

```r
print(attributes(bank$age))  # NULL means no special attributes defined
```

```
## NULL
```

```r
# plot a histogram for this variable
with(bank, hist(age))
```




```r
# 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")))
```

```
##
##        admin.   blue-collar  entrepreneur     housemaid    management
##           478           946           168           112           969
##       retired self-employed      services       student    technician
##           230           183           417            84           768
##    unemployed       unknown          <NA>
##           128            38             0
```

```r
print(table(bank$marital , useNA = c("always")))
```

```
##
## divorced  married   single     <NA>
##      528     2797     1196        0
```

```r
print(table(bank$education , useNA = c("always")))
```

```
##
##   primary secondary  tertiary   unknown      <NA>
##       678      2306      1350       187         0
```

```r
print(table(bank$default , useNA = c("always")))
```

```
##
##   no  yes <NA>
## 4445   76    0
```

```r
print(table(bank$housing , useNA = c("always")))
```

```
##
##   no  yes <NA>
## 1962 2559    0
```

```r
print(table(bank$loan , useNA = c("always")))
```

```
##
##   no  yes <NA>
## 3830  691    0
```

```r
# 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
```

```
##                jobtype
## job              White Collar Blue Collar Other/Unknown <NA>
##   admin.                  478           0             0    0
##   blue-collar               0         946             0    0
##   entrepreneur            168           0             0    0
##   housemaid                 0           0           112    0
##   management              969           0             0    0
##   retired                   0           0           230    0
##   self-employed           183           0             0    0
##   services                  0         417             0    0
##   student                   0           0            84    0
##   technician                0         768             0    0
##   unemployed                0           0           128    0
##   unknown                   0           0            38    0
##   <NA>                      0           0             0    0
```

```r
# 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))
```

```
## 'data.frame':    3705 obs. of  9 variables:
##  $ response : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 2 1 ...
##  $ age      : int  30 30 59 39 41 39 43 36 20 40 ...
##  $ jobtype  : Factor w/ 3 levels "White Collar",..: 3 1 2 2 1 2 1 2 3 1 ...
##  $ marital  : Factor w/ 3 levels "Divorced","Married",..: 2 2 2 2 2 2 2 2 3 2 ...
##  $ education: Factor w/ 4 levels "Primary","Secondary",..: 1 3 2 2 3 2 2 3 2 3 ...
##  $ default  : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ balance  : int  1787 1476 0 147 221 9374 264 1109 502 194 ...
##  $ housing  : Factor w/ 2 levels "No","Yes": 1 2 2 2 2 2 2 1 1 1 ...
##  $ loan     : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 1 1 1 1 2 ...
## NULL
```

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

```
##    response age       jobtype marital education default balance housing
## 1        No  30 Other/Unknown Married   Primary      No    1787      No
## 4        No  30  White Collar Married  Tertiary      No    1476     Yes
## 5        No  59   Blue Collar Married Secondary      No       0     Yes
## 8        No  39   Blue Collar Married Secondary      No     147     Yes
## 9        No  41  White Collar Married  Tertiary      No     221     Yes
## 11       No  39   Blue Collar Married Secondary      No    9374     Yes
##    loan
## 1    No
## 4   Yes
## 5    No
## 8    No
## 9    No
## 11   No
```

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

```
##  response        age                  jobtype         marital
##  No :3368   Min.   :19.00   White Collar :1453   Divorced: 443
##  Yes: 337   1st Qu.:33.00   Blue Collar  :1776   Married :2305
##             Median :39.00   Other/Unknown: 476   Single  : 957
##             Mean   :41.08
##             3rd Qu.:49.00
##             Max.   :87.00
##      education    default       balance      housing     loan
##  Primary  : 580   No :3634   Min.   :-3313   No :1662   No :3113
##  Secondary:1891   Yes:  71   1st Qu.:   60   Yes:2043   Yes: 592
##  Tertiary :1084              Median :  415
##  Unknown  : 150              Mean   : 1375
##                              3rd Qu.: 1412
##                              Max.   :71188
```

```
# 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()
```

```
## pdf
##   2
```

```
# 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"))))
```

```
##            response
## education     No  Yes <NA>
##   Primary    532   48    0
##   Secondary 1735  156    0
##   Tertiary   962  122    0
##   Unknown    139   11    0
##   <NA>         0    0    0
```

```
# 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"),
```

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

```
## pdf
##   2
```

```r
# job status using jobtype
# White Collar: admin., entrepreneur, management, self-employed
# Blue Collar: blue-collar, services, technician
# Other/Unknown
```

```r
# review the frequency table for job types
with(bankdata, print(table(jobtype, response, useNA = c("always"))))
```

```
##                  response
## jobtype             No  Yes <NA>
##   White Collar    1313  140    0
##   Blue Collar     1648  128    0
##   Other/Unknown    407   69    0
##   <NA>               0    0    0
```

```r
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()
```

```
## pdf
##   2
```

```r
# marital status
# Marital status (married, divorced, single)
# [Note: ``divorced'' means divorced or widowed]
```

```r
# 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"))))
```

```
##            response
## marital       No  Yes <NA>
##   Divorced   387   56    0
##   Married   2135  170    0
##   Single     846  111    0
##   <NA>         0    0    0
```

```r
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()
```

```
## pdf
##   2
```

```r
# default  Has credit in default? (yes, no)
```

```r
with(bankdata, print(table(default, response, useNA = c("always"))))
```

```
##         response
## default   No  Yes <NA>
##    No   3305  329    0
##    Yes    63    8    0
##    <NA>    0    0    0
```

```r
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()
```

```
## pdf
##   2
```

```r
# balance  Average yearly balance (in Euros)
```

```r
# 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()
```

```
## pdf
##   2
```

```r
# housing  Has housing loan? (yes, no)
```

```r
with(bankdata, print(table(housing, response, useNA = c("always"))))
```

```
##        response
## housing   No  Yes <NA>
##    No   1468  194    0
##    Yes  1900  143    0
##    <NA>    0    0    0
```

```r
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()
```

```
## pdf
##   2
```

```r
# loan  Has personal loan? (yes, no)
```

```r
with(bankdata, print(table(loan, response, useNA = c("always"))))
```

```
##       response
## loan     No  Yes <NA>
##   No   2806  307    0
##   Yes   562   30    0
##   <NA>    0    0    0
```

```r
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()
```

```
## pdf
##   2
```

```r
# specify predictive model
```

```r
bank_spec <- {response ~ age + jobtype + education + marital +
    default + balance + housing + loan}
```

```r
# fit logistic regression model
```

```r
bank_fit <- glm(bank_spec, family=binomial, data=bankdata)
print(summary(bank_fit))
```

```
##
## Call:
## glm(formula = bank_spec, family = binomial, data = bankdata)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.8546  -0.4787  -0.3985  -0.3247   2.7165
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -2.250e+00  4.072e-01  -5.526 3.27e-08 ***
## age                    1.004e-02  6.315e-03   1.591 0.111702
## jobtypeBlue Collar    -1.435e-01  1.447e-01  -0.992 0.321168
## jobtypeOther/Unknown   4.139e-01  1.771e-01   2.337 0.019443 *
## educationSecondary     1.036e-01  1.820e-01   0.569 0.569413
## educationTertiary      3.025e-01  2.043e-01   1.481 0.138716
## educationUnknown      -3.338e-01  3.527e-01  -0.946 0.344041
## maritalMarried        -5.717e-01  1.668e-01  -3.428 0.000608 ***
## maritalSingle         -3.509e-02  1.939e-01  -0.181 0.856376
## defaultYes             3.461e-01  3.876e-01   0.893 0.371917
## balance                4.783e-06  1.736e-05   0.276 0.782918
## housingYes            -4.058e-01  1.221e-01  -3.324 0.000888 ***
## loanYes               -6.961e-01  1.997e-01  -3.485 0.000491 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2258.2  on 3704  degrees of freedom
## Residual deviance: 2177.6  on 3692  degrees of freedom
## AIC: 2203.6
##
## Number of Fisher Scoring iterations: 5
```

```r
print(anova(bank_fit, test="Chisq"))
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: response
```

```
##
## Terms added sequentially (first to last)
##
##
##           Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                      3704     2258.2
## age        1   3.4257      3703     2254.8 0.0641901 .
## jobtype    2  20.1014      3701     2234.7 4.316e-05 ***
## education  3   8.0101      3698     2226.7 0.0458042 *
## marital    2  23.4978      3696     2203.2 7.898e-06 ***
## default    1   0.2848      3695     2202.9 0.5935650
## balance    1   0.2644      3694     2202.6 0.6071299
## housing    1  10.7676      3693     2191.8 0.0010329 **
## loan       1  14.2114      3692     2177.6 0.0001634 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# 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()
```

```
## pdf
##   2
```

```
# 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")
```

```
##
## Confusion Matrix (rows=Predicted Response, columns=Actual Choice
```

```
print(confusion_matrix)
```

```
##
```

```
##         No  Yes
##   NO  3368  337
##   YES    0    0
```

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

```
##
## Percent Accuracy:  90.9
```

```r
# 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")
```

```
##
## Confusion Matrix (rows=Predicted Response, columns=Actual Choice
```

```r
print(confusion_matrix)
```

```
##
##         No  Yes
##   NO  1651  102
##   YES 1717  235
```

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

```
##
## Percent Accuracy:  50.9
```

```r
# mosaic rendering of the classifier with 0.10 cutoff
with(bankdata, print(table(Predict_Response, response, useNA = c("always"))))
```

```
##                 response
## Predict_Response   No  Yes <NA>
##             NO   1651  102    0
##             YES  1717  235    0
##             <NA>    0    0    0
```

```r
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 =
```

```r
  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()
```

```
## pdf
##   2
```

```r
# 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()
```

```
## pdf
##   2
```

```r
# 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")
```

```
##
## Lift Chart Values by Decile:
```

```r
print(by(bankdata$response_binary, bankdata$decile,
    function(x) lift(x, baseline_response_rate)))
```

```
## bankdata$decile: Decile_10
## [1] 0.4741376
## ---------------------------------------------------------
## bankdata$decile: Decile_9
## [1] 0.5348464
## ---------------------------------------------------------
## bankdata$decile: Decile_8
## [1] 0.592672
## ---------------------------------------------------------
## bankdata$decile: Decile_7
## [1] 0.8022696
## ---------------------------------------------------------
## bankdata$decile: Decile_6
## [1] 0.8593744
## ---------------------------------------------------------
## bankdata$decile: Decile_5
## [1] 0.861697
## ---------------------------------------------------------
## bankdata$decile: Decile_4
## [1] 1.218261
## ---------------------------------------------------------
## bankdata$decile: Decile_3
## [1] 1.303878
## ---------------------------------------------------------
## bankdata$decile: Decile_2
## [1] 1.12912
## ---------------------------------------------------------
## bankdata$decile: Decile_1
## [1] 2.22252
```

```r
# 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.
```