PA Day-03

Workshop: 01

Implement Your Logistic Regression Model in R

Data file: ~~marketing.xlsx~~ bank_data.csv

Meta-data

1. Refer to the dataset marketing.xls for data and meta-data information.

Case: A Portuguese bank conducted seventeen telephone marketing campaigns between May 2008 and November 2010. The bank recorded client contact information for each telephone call. The bank wants its clients to invest in term deposits. A term deposit is an investment such as a certificate of deposit. The interest rate and duration of the deposit are set in advance. A term deposit is distinct from a demand deposit. The bank is interested in identifying factors that affect client responses to new term deposit offerings, which are the focus of the marketing campaigns.

Dataset: Client characteristics include demographic factors: age, job type, marital status, and education. The client's previous use of banking services is also noted. Current contact information shows the date of the telephone call and the duration of the call. There is also information about the call immediately preceding the current call, as well as summary information about all calls with the client.

The classification goal is to predict if the client will subscribe (yes/no) to a term deposit (variable y). Analyze the data using logistic regression technique. Submit a one-page report covering:

1. What are the important determinants of a positive subscription (Y/N)
2. How you tuned the model
3. Use of lift chart for better targeting

DataSource: https://archive.ics.uci.edu/ml/datasets/Bank+Marketing# (https://archive.ics.uci.edu/ml/datasets/Bank+Marketing#)

A. Bank client data:

1. - age (numeric)
2. - job: type of job (categorical: 'admin' ,'blue-collar', 'entrepreneur', 'housemaid', 'management', 'retired','self-employed','services','student','technician','unemployed','unknown')
3. - marital : marital status (categorical: 'divorced', 'married', 'single', 'unknown'; note: 'divorced' means divorced or widowed)
4. - education (categorical: 'basic.4y', 'basic.6y', 'basic.9y', 'high school', 'illiterate', 'professional course', 'university degree', 'unknown')
5. - default: has credit in default? (categorical: 'no', 'yes', 'unknown')
6. - housing: has housing loan? (categorical: 'no', 'yes', 'unknown')
7. - loan: has personal loan? (categorical: 'no', 'yes', 'unknown')

B. Related to the last contact of the current campaign:

1. - contact: contact communication type (categorical: 'cellular', 'telephone')
2. - month: last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec')
3. - day_of_week: last contact day of the week (categorical: 'mon','tue','wed','thu','fri')
4. - duration: last contact duration, in seconds (numeric). Important note: this attribute highly affects the output target (e.g., if duration=0 then y='no'). Yet, the duration is not known before a call is performed. Also, after the end of the call y is obviously known. Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model.

C. Other attributes:

1. - campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)
2. - pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)
3. - previous: number of contacts performed before this campaign and for this client (numeric)
4. - poutcome: outcome of the previous marketing campaign (categorical: 'failure', 'non-existent', 'success')

E. Output variable (desired target)

1. - y - has the client subscribed a term deposit? (binary: 'yes','no')

In [1]:

```
# let's prepare our workspace
pacman::p_load(tidyverse, caret, corrplot, caTools,car, ROCR)
```

In [2]:

```
# path to data file: marketing.csv
# pay attention to the '/'
setwd("C:/Users/isspcs/Desktop/workshop-data")
```

Step-1: Load Your Data using R

In [3]:

```
data =  read.csv("bank_data.csv")
```

Step-2: Explore Your Data

In [5]:

```
# structure of our data
str(data)
```

```
'data.frame':	45211 obs. of  17 variables:
 $ age      : int  58 44 33 47 33 35 28 42 58 43 ...
 $ job      : Factor w/ 12 levels "admin.","blue-collar",..:
5 10 3 2 12 5 5 3 6 10 ...
 $ marital  : Factor w/ 3 levels "divorced","married",..: 2 3
2 2 3 2 3 1 2 3 ...
 $ education: Factor w/ 4 levels "primary","secondary",..: 3
2 2 4 4 3 3 3 1 2 ...
 $ default  : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 2
1 1 ...
 $ balance  : int  2143 29 2 1506 1 231 447 2 121 593 ...
 $ housing  : Factor w/ 2 levels "no","yes": 2 2 2 2 1 2 2 2
2 2 ...
 $ loan     : Factor w/ 2 levels "no","yes": 1 1 2 1 1 1 2 1
1 1 ...
 $ contact  : Factor w/ 3 levels "cellular","telephone",..: 3
3 3 3 3 3 3 3 3 3 ...
 $ day      : int  5 5 5 5 5 5 5 5 5 5 ...
 $ month    : Factor w/ 12 levels "apr","aug","dec",..: 9 9 9
9 9 9 9 9 9 9 ...
 $ duration : int  261 151 76 92 198 139 217 380 50 55 ...
 $ campaign : int  1 1 1 1 1 1 1 1 1 1 ...
 $ pdays    : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
 $ previous : int  0 0 0 0 0 0 0 0 0 0 ...
 $ poutcome : Factor w/ 4 levels "failure","other",..: 4 4 4
4 4 4 4 4 4 4 ...
 $ y        : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1
1 1 ...
```

In [6]:

```
# see how head command can be used
head(data, 4)
```

age	job	marital	education	default	balance	housing	loan
58	management	married	tertiary	no	2143	yes	no
44	technician	single	secondary	no	29	yes	no
33	entrepreneur	married	secondary	no	2	yes	yes
47	blue-collar	married	unknown	no	1506	yes	no

In [7]:

```
options(repr.plot.width=6, repr.plot.height=3)
theme_set(theme_bw())
```

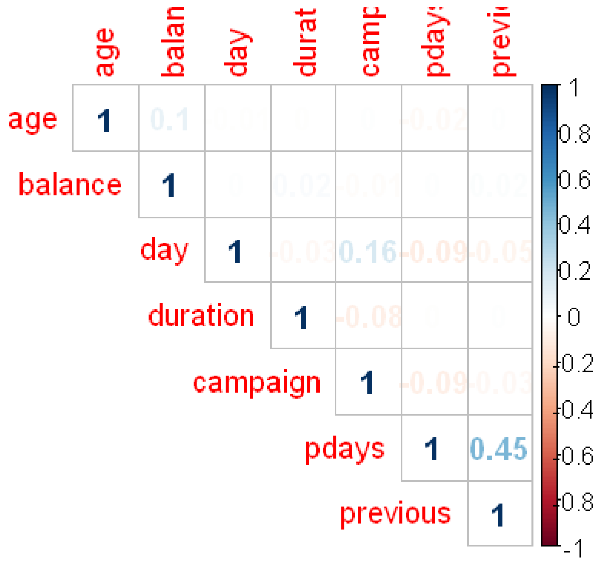
In [8]:

```
#renaming some variables
data = data %>%
rename(response = y)

head(data,2)
```

age	job	marital	education	default	balance	housing	loan
58	management	married	tertiary	no	2143	yes	no
44	technician	single	secondary	no	29	yes	no

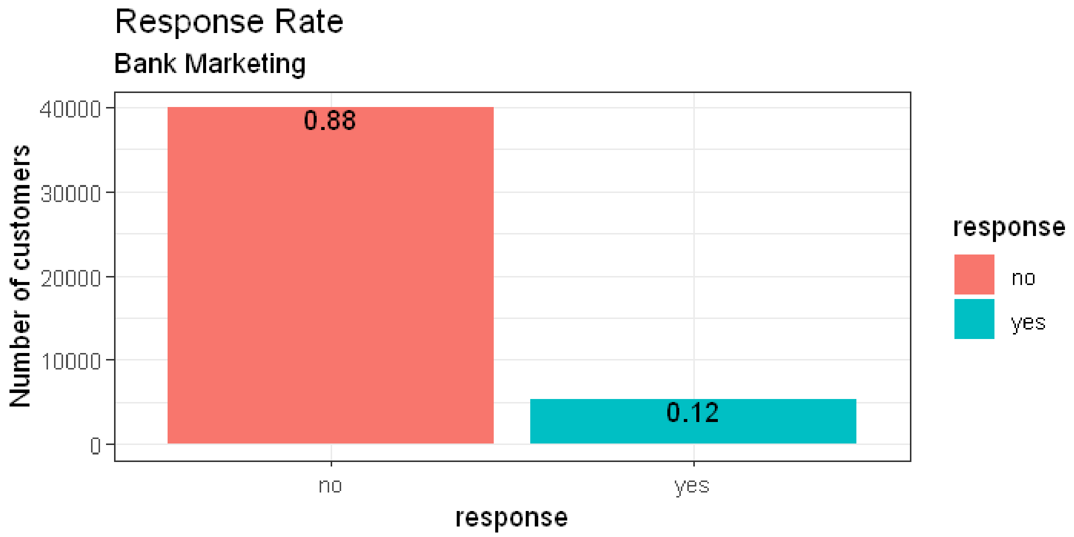
In [9]:

```
corrplot(cor(data[sapply(data, is.numeric)]), method = "number", type='upp
er')
```



In [10]:

```
data %>%
    group_by(response) %>%
    summarise(count_level = n(),
    percentage = n()/nrow(data))%>%
    ggplot(aes(x = response,
               y = count_level,fill=response )) +
    geom_bar(stat='identity') +
    geom_text(aes(label=round(percentage,2)),vjust = 1)+
    labs(x= "response", y= "Number of customers",
         title = "Response Rate",
         subtitle = "Bank Marketing")
```



Observations

1. One of the challenges in target marketing is to with low rates in response to promotional efforts.
2. Here we can see that only 12% of 4521 bank clients responded favourably to bank's offer.

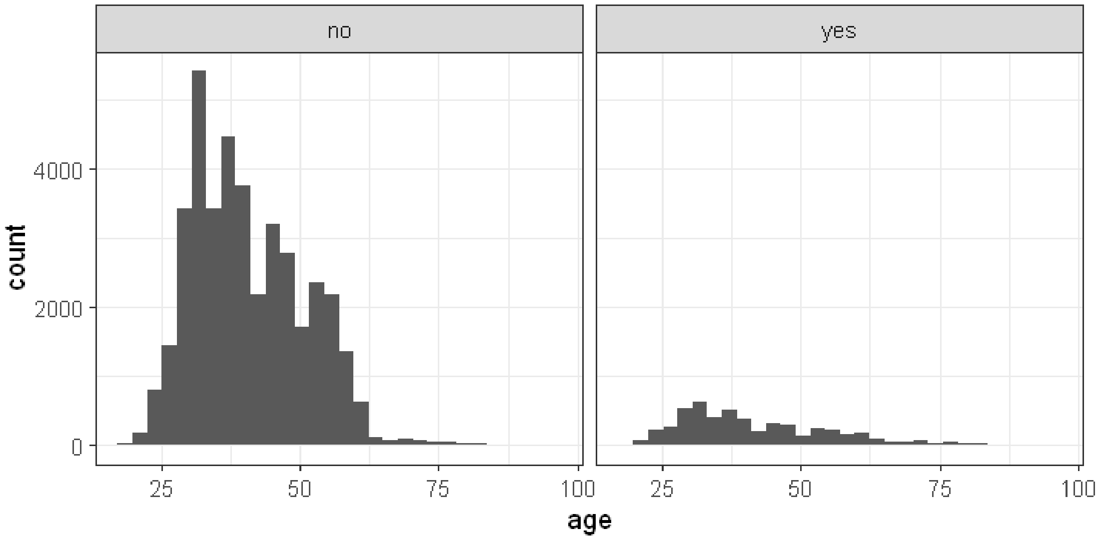
Do demographics play a part?

1. Let's see if there is any relation (visual exploration)
2. What are demographic variables avaialble to us?

In [11]:

```
data %>%
ggplot(aes(age))+ geom_histogram() + facet_grid(~ response)
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwi
dth`.
```



education

In [12]:

```
tb = round(prop.table(table(data$response, data$education), 2)*100,2)
tb
```

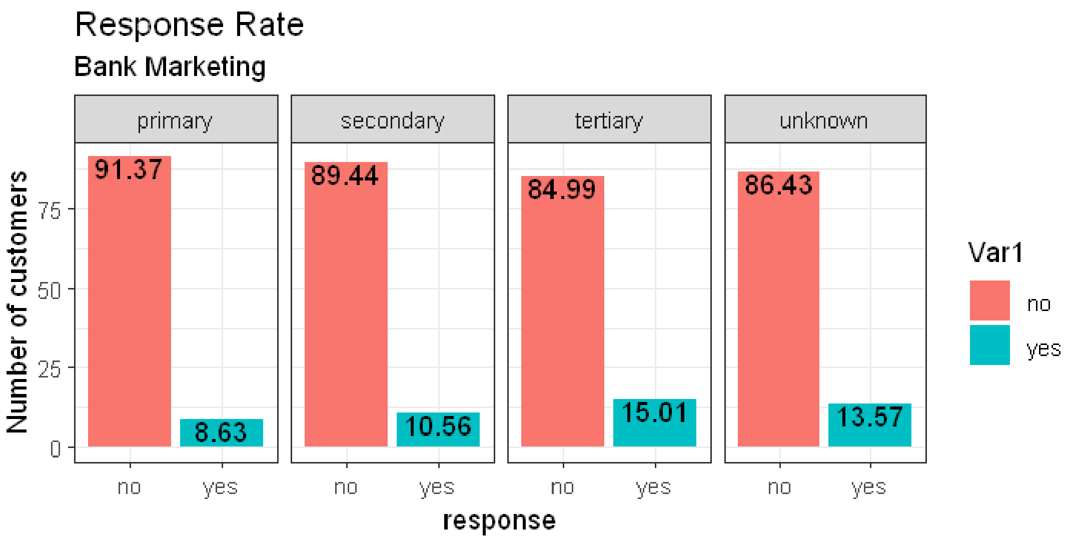
```
      primary secondary tertiary unknown
  no    91.37     89.44    84.99   86.43
  yes    8.63     10.56    15.01   13.57
```

In [13]:

```
# barplot(tb, main="Distribution by Education",
#   xlab="Education Level", col=c("red","blue"),
#   legend = rownames(counts), beside=TRUE)
```

In [14]:

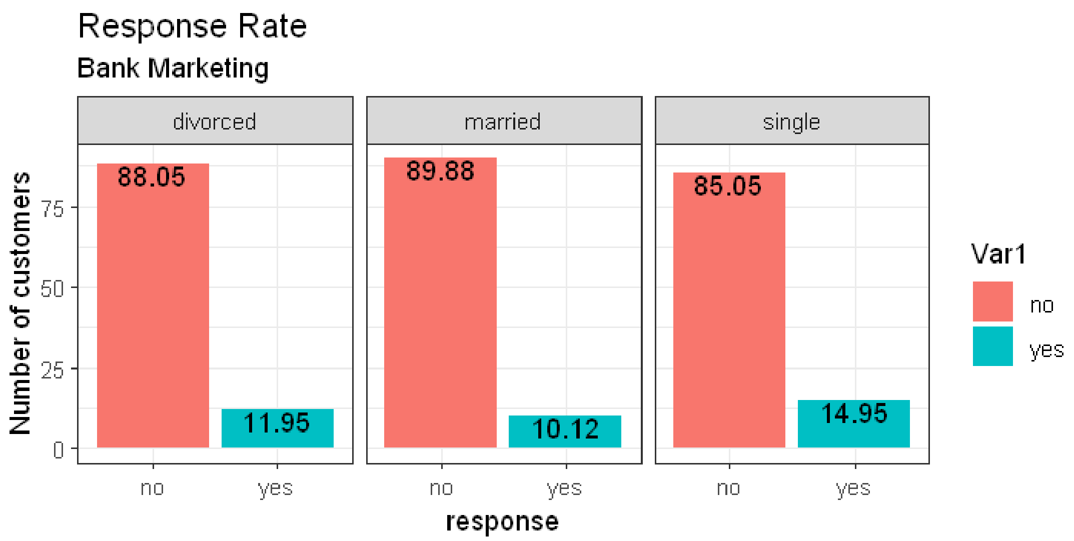
```
data.frame(tb) %>%
ggplot(aes(x= Var1, y = Freq, fill= Var1)) +geom_bar(stat="identity") + fa
cet_grid(~Var2) +
    geom_text(aes(label=round(Freq,2)),vjust = 1)+
    labs(x= "response", y= "Number of customers",
         title = "Response Rate",
         subtitle = "Bank Marketing")
```



marital status

In [15]:

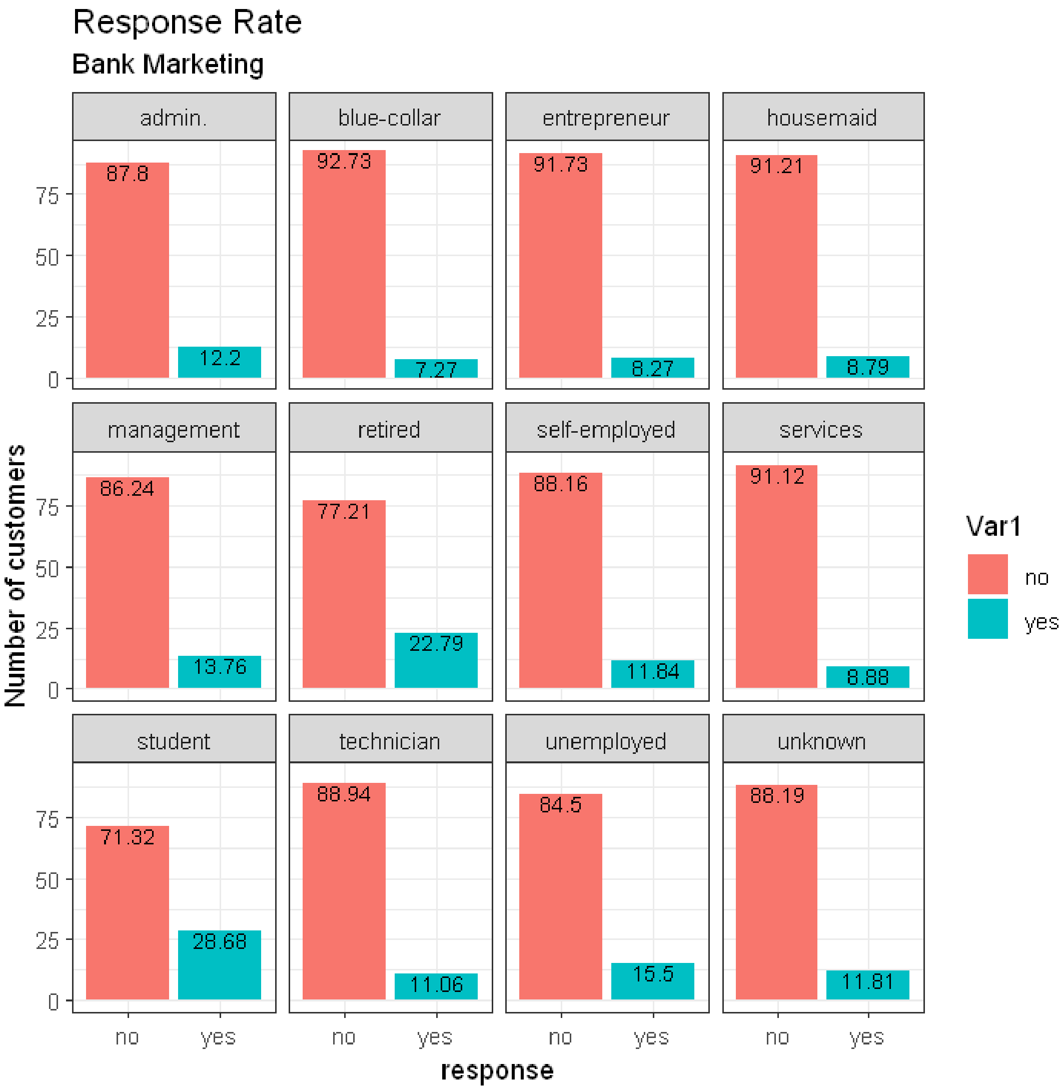
```
tb = round(prop.table(table(data$response, data$marital), 2)*100,2)

data.frame(tb) %>%
ggplot(aes(x= Var1, y = Freq, fill= Var1)) +geom_bar(stat="identity") + fa
cet_grid(~Var2) +
    geom_text(aes(label=round(Freq,2)),vjust = 1)+
    labs(x= "response", y= "Number of customers",
         title = "Response Rate",
         subtitle = "Bank Marketing")
```



job type

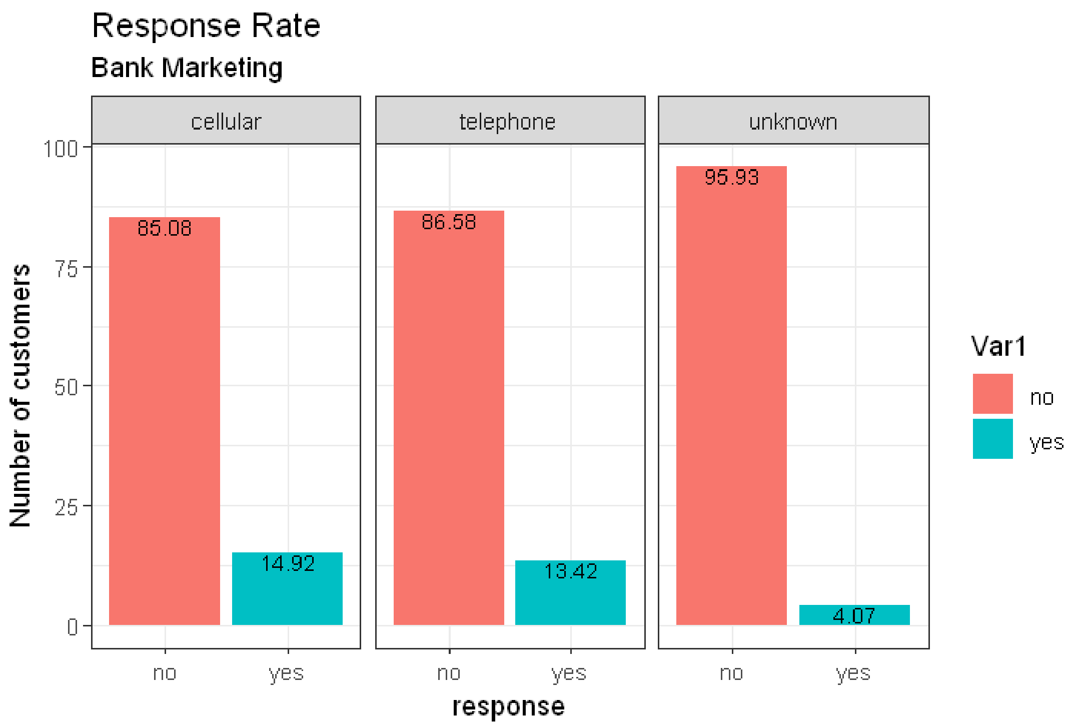
In [16]:

```
options(repr.plot.width=6, repr.plot.height=6)
tb = round(prop.table(table(data$response, data$job), 2)*100,2)

data.frame(tb) %>%
ggplot(aes(x= Var1, y = Freq, fill= Var1)) +geom_bar(stat="identity") + fa
cet_wrap(~Var2) +
    geom_text(aes(label=round(Freq,2)),vjust = 1, size =3)+
    labs(x= "response", y= "Number of customers",
         title = "Response Rate",
         subtitle = "Bank Marketing")
```



In [18]:

```
options(repr.plot.width=6, repr.plot.height=4)
tb = round(prop.table(table(data$response, data$contact), 2)*100,2)

data.frame(tb) %>%
ggplot(aes(x= Var1, y = Freq, fill= Var1)) +geom_bar(stat="identity") + fa
cet_wrap(~Var2) +
    geom_text(aes(label=round(Freq,2)),vjust = 1, size =3)+
    labs(x= "response", y= "Number of customers",
         title = "Response Rate",
         subtitle = "Bank Marketing")
```



Observations

1. more likely to be older, more highly educated, white collar jobs such as admin, management, retired, self-employed or students
2. more likely to be divorced or single.
3. more likely to be contacted via cellular or telephone

```
Question: how about loan v/s response ? (left as an exercise for st
udents)
```

Model Building

Note :

When you are developing a propensity model from campaign data keep in mind that your explanatory variables will be customer features and not campaign features. Please remember about the implementation and future usage. When you are going to use this model in future for a new campaign, all you are going to do is to find their propensity to buy (either in the form of probability or the linear component of the logistic regression which is often termed as score) and select prospective customers with high probability or high score. The data takes care of seasonality as it has data from May 2008 to November 2010. (In the data set these months May, Jun…. are referred to as scoring month).

The following variables, can be selected as initial candidate:

1. Group A

   1 - age 2 - job: 3 - marital : 4 - education 5 - default: 6 - housing: 7 - loan: balance

   It is term deposit cross-sell so it is ok to use default as input variable but if it were some loan cross-sell then you will be deleting defaulted customers from your development base as they are exclusion, because you don't want to cross-sell another loan to a defaulted customer.

1. Group B

   8 – contact. Rest of the variables of this group will not be considered as they are campaign characteristics. For variable 11 explanation is given why it should not be used.

1. Group C

   12 – campaign should not be used as for the next campaign at the beginning, this will not be available so can't be used in scorecard. Obviously typically more number of times you contact, higher is the chance of conversion. 13 - pdays: 14 – previous: 15 - poutcome:

1. Group D

   These variables shouldn't be used either as they are macroeconomic variables and not customer characteristics. So they remain constant for all customers at any given time point. However, you will see that they are influencing campaign success because they reflect boom time or recession. But they are not customer characteristic (constant for everybody at any given time point).

In [19]:

```
#colnames(data)
```

In [20]:

```
contrasts(data$response)
```

	yes
no	0
yes	1

split data into train and test

In [21]:

```
#set initial seed
set.seed(123)

# create a boolean flag to split data
splitData = sample.split(data$response, SplitRatio = 0.7)

#split_data
# create train and test datasets
train_set = data[splitData,]

nrow(train_set)/nrow(data)

test_set = data[!splitData,]
nrow(test_set)/nrow(data)
```

0.699984517042313

0.300015482957687

In [22]:

```
model1 = glm(response ~ age+ job+marital+ education+default+ balance+ hous
ing+
                loan+ contact+ pdays+ previous+poutcome ,
             data = train_set, family = binomial)
summary(model1)
```

```
Call:
glm(formula = response ~ age + job + marital + education + de
fault +
    balance + housing + loan + contact + pdays + previous + p
outcome,
    family = binomial, data = train_set)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9586  -0.5105  -0.3960  -0.2641   2.9110

Coefficients:
                     Estimate Std. Error z value Pr(>|z|)
(Intercept)        -1.765e+00  1.768e-01  -9.980  < 2e-16 ***
age                 2.493e-03  2.293e-03   1.087 0.276952
jobblue-collar     -1.668e-01  7.584e-02  -2.199 0.027861 *
jobentrepreneur    -2.635e-01  1.288e-01  -2.045 0.040882 *
jobhousemaid       -4.936e-01  1.483e-01  -3.327 0.000876 ***
jobmanagement      -1.114e-01  7.694e-02  -1.448 0.147696
jobretired          5.185e-01  9.911e-02   5.232 1.68e-07 ***
jobself-employed   -7.909e-02  1.139e-01  -0.695 0.487351
jobservices        -1.437e-01  8.789e-02  -1.635 0.102007
jobstudent          4.863e-01  1.167e-01   4.166 3.10e-05 ***
jobtechnician      -1.826e-01  7.250e-02  -2.518 0.011789 *
jobunemployed       1.156e-01  1.144e-01   1.011 0.312117
jobunknown         -2.043e-02  2.285e-01  -0.089 0.928752
maritalmarried     -1.291e-01  6.155e-02  -2.098 0.035901 *
maritalsingle       1.794e-01  7.029e-02   2.552 0.010700 *
educationsecondary  1.667e-01  6.684e-02   2.494 0.012628 *
educationtertiary   3.351e-01  7.784e-02   4.305 1.67e-05 ***
educationunknown    2.635e-01  1.077e-01   2.447 0.014422 *
defaultyes         -2.889e-01  1.794e-01  -1.610 0.107306
balance             2.255e-05  5.002e-06   4.508 6.54e-06 ***
housingyes         -5.686e-01  4.177e-02 -13.613  < 2e-16 ***
loanyes            -4.717e-01  6.101e-02  -7.732 1.06e-14 ***
contacttelephone   -2.444e-01  7.511e-02  -3.254 0.001139 **
contactunknown     -9.728e-01  6.043e-02 -16.099  < 2e-16 ***
pdays               1.606e-04  3.202e-04   0.502 0.615908
previous            3.346e-03  6.435e-03   0.520 0.603022
poutcomeother       2.953e-01  9.137e-02   3.231 0.001232 **
poutcomesuccess     2.212e+00  8.645e-02  25.585  < 2e-16 ***
poutcomeunknown    -1.967e-01  9.645e-02  -2.039 0.041441 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 22840  on 31646  degrees of freedom
Residual deviance: 19796  on 31618  degrees of freedom
AIC: 19854
```

```
Number of Fisher Scoring iterations: 6
```

In [23]:

```
vif(model1)
```

	GVIF	Df	GVIF^(1/(2*Df))
age	2.086294	1	1.444401
job	3.711689	11	1.061426
marital	1.421401	2	1.091891
education	2.249884	3	1.144704
default	1.009829	1	1.004903
balance	1.032032	1	1.015890
housing	1.196942	1	1.094049
loan	1.028364	1	1.014083
contact	1.170433	2	1.040128
pdays	3.750867	1	1.936716
previous	1.254572	1	1.120077
poutcome	4.127583	3	1.266531

In [24]:

```
# test it on the train set
trainPredict = predict(model1, newdata = train_set, type = 'response')
p_class = ifelse(trainPredict > 0.5, 'yes','no')

matrix_table = table(train_set$response, p_class)
matrix_table
```

```
     p_class
         no   yes
  no  27623   322
  yes  3081   621
```

In [25]:

```
# Accuracy
accuracy = sum(diag(matrix_table))/sum(matrix_table)
round(accuracy, 3)*100
```

89.2

2nd iteration :

```
Drop age, pdays, previous and default  and run the model again. It
 doesn’t improve the result.
```

3rd iteration :

```
We drop Job as many categories are insignificant.
```

In [26]:

```
model3 = glm(response ~ marital+ education+ balance+ housing+
                loan+ contact+ poutcome ,
             data = train_set, family = binomial)
summary(model3)
```

```
Call:
glm(formula = response ~ marital + education + balance + hous
ing +
    loan + contact + poutcome, family = binomial, data = trai
n_set)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8278  -0.5246  -0.3979  -0.2652   2.8777

Coefficients:
                     Estimate Std. Error z value Pr(>|z|)
(Intercept)        -1.557e+00  9.384e-02 -16.591  < 2e-16 ***
maritalmarried     -1.644e-01  6.085e-02  -2.701  0.00691 **
maritalsingle       1.411e-01  6.417e-02   2.199  0.02786 *
educationsecondary  1.257e-01  6.150e-02   2.044  0.04099 *
educationtertiary   2.590e-01  6.431e-02   4.028 5.63e-05 ***
educationunknown    2.938e-01  1.030e-01   2.852  0.00434 **
balance             2.542e-05  4.942e-06   5.145 2.68e-07 ***
housingyes         -6.506e-01  3.967e-02 -16.399  < 2e-16 ***
loanyes            -5.068e-01  6.060e-02  -8.363  < 2e-16 ***
contacttelephone   -1.508e-01  7.285e-02  -2.069  0.03851 *
contactunknown     -9.789e-01  6.022e-02 -16.256  < 2e-16 ***
poutcomeother       3.020e-01  9.044e-02   3.339  0.00084 ***
poutcomesuccess     2.226e+00  8.330e-02  26.722  < 2e-16 ***
poutcomeunknown    -2.705e-01  5.774e-02  -4.684 2.81e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 22840  on 31646  degrees of freedom
Residual deviance: 19937  on 31633  degrees of freedom
AIC: 19965

Number of Fisher Scoring iterations: 6
```

In [27]:

```
# test it on the train set
trainPredict = predict(model3, newdata = train_set, type = 'response')
p_class = ifelse(trainPredict > 0.5, 'yes','no')

matrix_table = table(train_set$response, p_class)
matrix_table
```

```
     p_class
         no   yes
  no  27607   338
  yes  3066   636
```

In [28]:

```
# Accuracy
accuracy = sum(diag(matrix_table))/sum(matrix_table)
round(accuracy, 3)*100
```

89.2

4th iteration:

```
There is slight improvement. However the last variable balance, eve
n though it is significant, the corresponding co-efficient is very
 small. So we drop this variable and run the algorithm again.
 We drop balance.
```

In [29]:

```
model4 = glm(response ~ marital+ education+ housing+
                loan+ contact+ poutcome ,
             data = train_set, family = binomial)
summary(model4)
```

```
Call:
glm(formula = response ~ marital + education + housing + loan
+
    contact + poutcome, family = binomial, data = train_set)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6877  -0.5265  -0.4001  -0.2673   2.8742

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)        -1.51704    0.09345 -16.233  < 2e-16 ***
maritalmarried     -0.15642    0.06077  -2.574 0.010059 *
maritalsingle       0.14065    0.06413   2.193 0.028283 *
educationsecondary  0.12455    0.06146   2.027 0.042704 *
educationtertiary   0.27370    0.06420   4.263 2.01e-05 ***
educationunknown    0.30050    0.10293   2.920 0.003505 **
housingyes         -0.66013    0.03962 -16.662  < 2e-16 ***
loanyes            -0.52671    0.06045  -8.713  < 2e-16 ***
contacttelephone   -0.13399    0.07257  -1.846 0.064848 .
contactunknown     -0.97828    0.06021 -16.249  < 2e-16 ***
poutcomeother       0.30307    0.09035   3.354 0.000795 ***
poutcomesuccess     2.22472    0.08325  26.722  < 2e-16 ***
poutcomeunknown    -0.27573    0.05771  -4.778 1.77e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 22840  on 31646  degrees of freedom
Residual deviance: 19962  on 31634  degrees of freedom
AIC: 19988

Number of Fisher Scoring iterations: 6
```

In [30]:

```
# test it on the train set
trainPredict = predict(model4, newdata = train_set, type = 'response')
p_class = ifelse(trainPredict > 0.3, 'yes','no')

matrix_table = table(train_set$response, p_class)
matrix_table
```

```
     p_class
         no   yes
  no  27526   419
  yes  2991   711
```

In [31]:

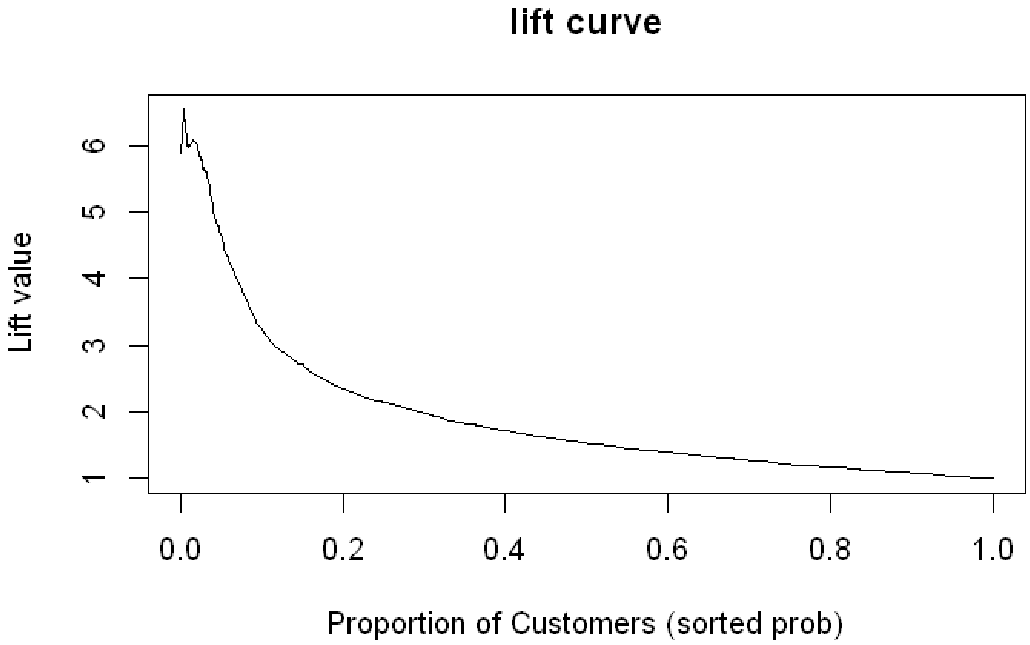
```
# Accuracy
accuracy = sum(diag(matrix_table))/sum(matrix_table)
round(accuracy, 3)*100
```

89.2

```
Accuracy is almost same from the first model : 89.3% compared to th
e first run, however the type II error is lower.

So we stop here and note that all the variables are significant. Th
ough one category of marital & education is not significant but ove
rall they are significant.  With existing choice of variables we ca
n’t make further improvement. Typically in such situation when we h
ave one group much bigger than the other (39922 vs 5289) the larger
group dominates and that is reflected in the model which is the cas
e here. Model is biased towards ‘no’ group. To eliminate this impac
t many a times what is done is no. in larger group is reduced e.g.
 taking a random sample of say 1000 of ‘no’ and all 521 of ‘yes’ an
d then build the model. This is known as over/under sampling ( Stat
istics) or data balancing ( Machine Learning).
```

In [32]:

```
pred = prediction( trainPredict, train_set$response )

perf = performance( pred, "lift", "rpp" )
plot(perf, main="lift curve", xlab = 'Proportion of Customers (sorted pro
b)')
```



test on test_set

In [35]:

```
# test it on the train set
testPredict = predict(model4, newdata = test_set, type = 'response')
p_class = ifelse(testPredict > 0.5, 'yes','no')

matrix_table = table(test_set$response, p_class)
matrix_table

# Accuracy
accuracy = sum(diag(matrix_table))/sum(matrix_table)
round(accuracy, 3)*100
```

```
     p_class
         no   yes
  no  11840   137
  yes  1307   280
```

89.4

Accuracy on test data is comparable to train data