Exercise 7 - Linear and Logisitic Regression using R

S.DEIVANAYKI (21BCS003)

12.03.2024

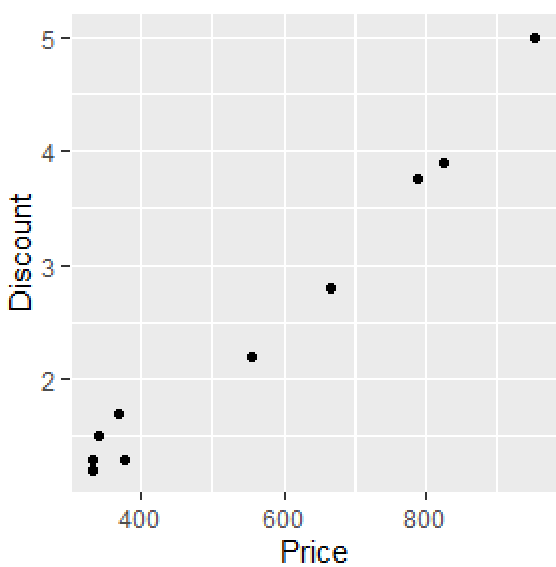
#Create a DataFrame with Columns as Price and Discount.

```
df<-
data.frame(Price=c(368,340,665,954,331,556,376,332,788,826),Discount=c(1.7,1.
5,2.8,5,1.3,2.2,1.3,1.2,3.75,3.9))
head(df)
```

```
##   Price Discount
## 1   368      1.7
## 2   340      1.5
## 3   665      2.8
## 4   954      5.0
## 5   331      1.3
## 6   556      2.2
```

#Visualize the Dataframe

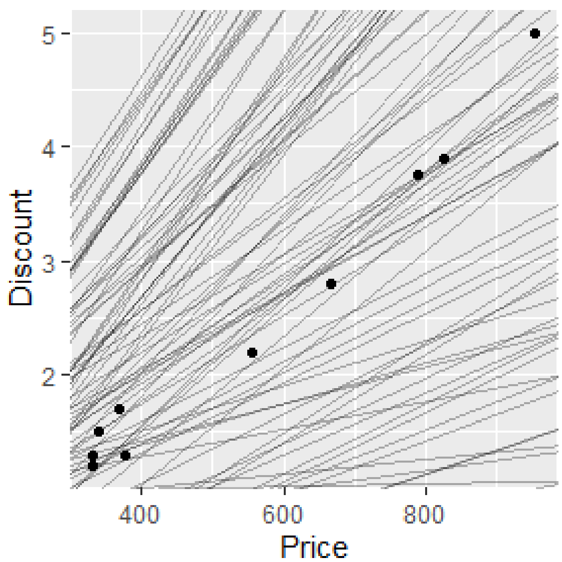
```
library(ggplot2)
ggplot(df,aes(Price,Discount))+geom_point()
```



#1. Create models using geom_abline().

```
library(tidyverse)
```

```r
models<-tibble(a1=runif(200,-1,1),a2=runif(200,-0.01,0.01))
ggplot(df,aes(Price,Discount))+geom_abline(aes(intercept=a1,slope=a2),data=mo
dels,alpha=0.3)+geom_point()
```

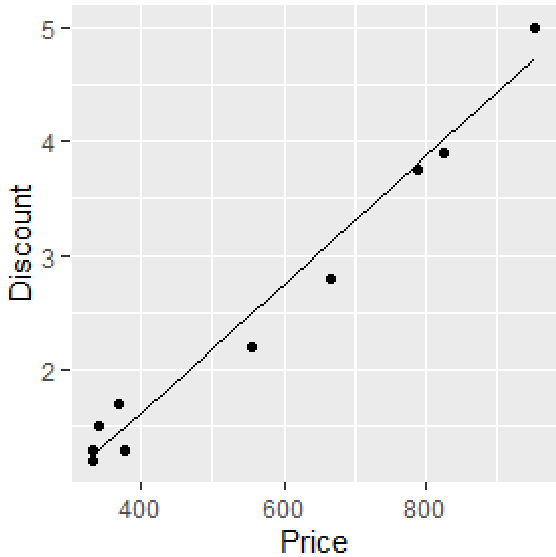


#Create model for specific slope and intercept

```r
model1 <- function(a,data)
{
  a[1]+data$Price*a[2]
}
y_pred <- model1(c(-0.612,0.0056),df)
y_pred

##  [1] 1.4488 1.2920 3.1120 4.7304 1.2416 2.5016 1.4936 1.2472 3.8008 4.0136
```

#Visualize Actual Vs Predicted Discount

```r
ggplot(df,aes(Price))+geom_point(aes(y=Discount))+geom_line(aes(y=y_pred))
```



#Calculate Root mean square error

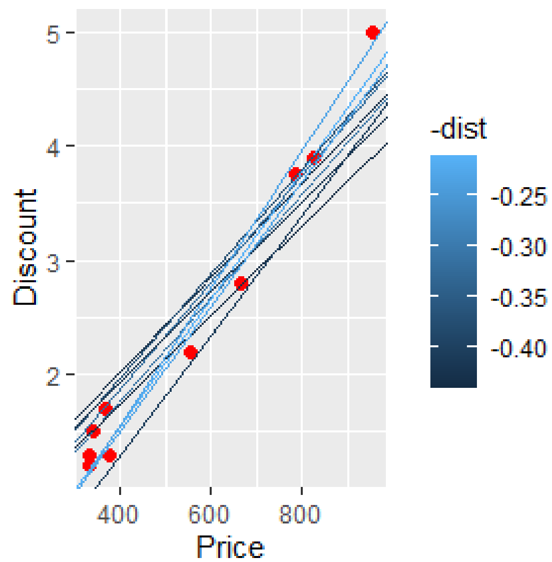
```
measure_distance <- function(mod,data)
{
  diff<-df$Discount-model1(mod,data)
  sqrt(mean(diff^2))
}
rmse <- measure_distance(c(-0.612,0.0056),df)
rmse

## [1] 0.2063883
```

#2.Generate intercept and slope using purrr

```
library(purrr)
xy_dist <- function(a1,a2)
{
  measure_distance(c(a1,a2),df)
}

models <- models %>% mutate(dist = purrr::map2_dbl(a1,a2,xy_dist))
head(models)

## # A tibble: 6 × 3
##         a1       a2  dist
##      <dbl>    <dbl> <dbl>
## 1 -0.144  -0.00266 4.49
## 2 -0.0110  0.00450 0.313
## 3  0.968  -0.00817 6.77
## 4 -0.298   0.00750 1.47
## 5  0.310   0.00324 0.666
## 6  0.252   0.00969 3.29
```

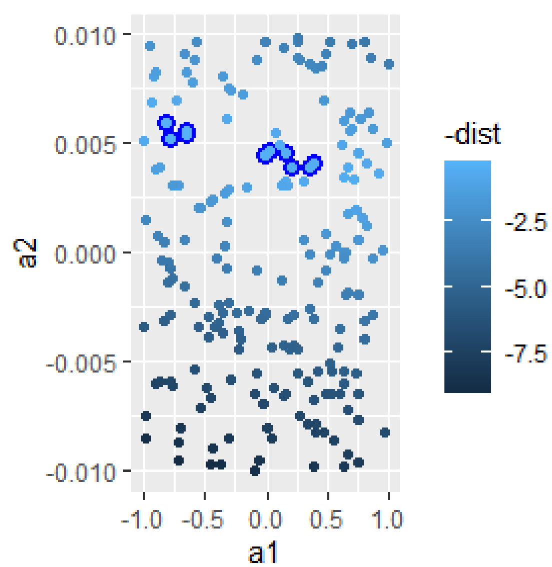
#Visualise the top 10 models

```r
ggplot(df,aes(Price,Discount))+geom_point(size=2,color="red")+geom_abline(aes
(intercept=a1,slope=a2,color=-dist),data=filter(models,rank(dist)<=10))
```



#Visualize the top 10 intercept and slope.

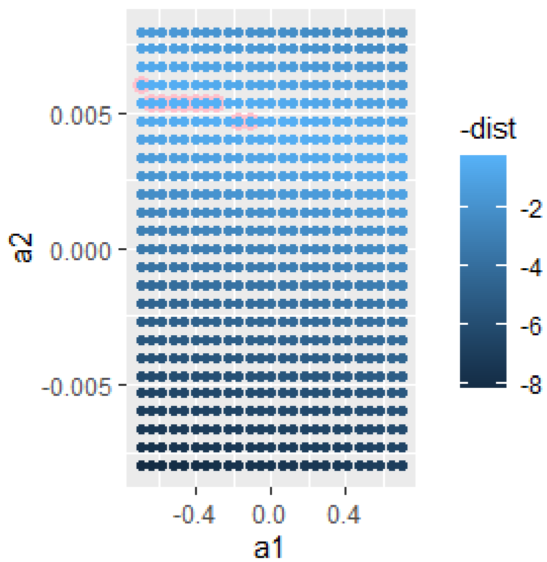
```r
ggplot(models,aes(a1,a2))+geom_point(data=filter(models,rank(dist)<=10),size=
3,color="blue")+geom_point(aes(colour=-dist))
```



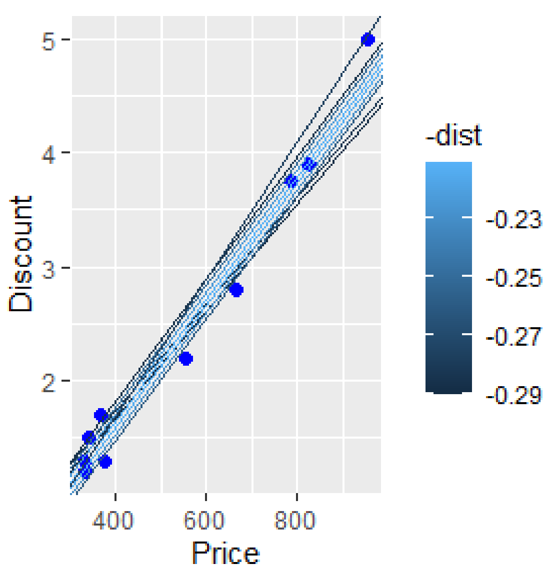
#3.Visualize the Best Models using Grid Search.

```r
grid <- expand.grid(a1=seq(-0.7,0.7,length=25),a2=seq(-
0.008,0.008,length=25)) %>% mutate(dist = purrr::map2_dbl(a1,a2,xy_dist))

grid %>%
```

```r
ggplot(aes(a1,a2))+geom_point(data=filter(grid,rank(dist)<=10),size=3,color="
pink")+geom_point(aes(color=-dist))
```

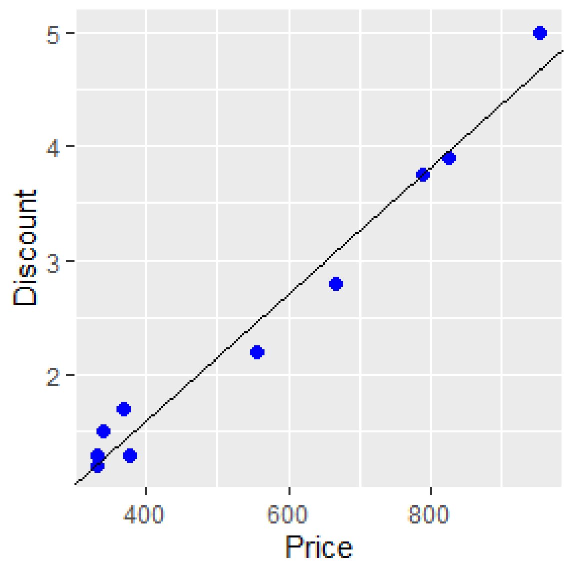



```r
ggplot(df,aes(Price,Discount)) + geom_point(size=2,color="blue") +
geom_abline(aes(intercept=a1,slope=a2,color=-
dist),data=filter(grid,rank(dist)<=10))
```



#4. Newton-Raphson Method to find slope and intercept

```r
best <- optim(c(0,0),measure_distance,data=df)
best$par

## [1] -0.603489635  0.005542682

ggplot(df,aes(Price,Discount)) + geom_point(size=2,color="blue") +
geom_abline(intercept=best$par[1],slope=best$par[2])
```



#5. Create a linear model using lm().

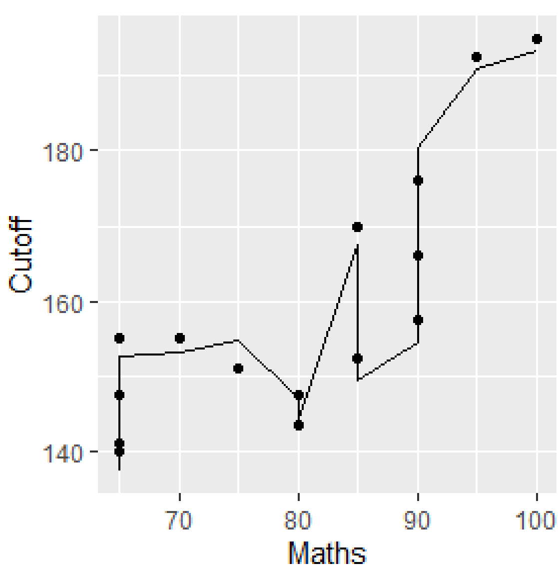
```
ind=sample(1:10,7)
data_train <- df[ind,]
data_test <- df[-ind,]
linear_model <- lm(Discount~Price,data=data_train)
summary(linear_model)

##
## Call:
## lm(formula = Discount ~ Price, data = data_train)
##
## Residuals:
##        1       10        4        3        8        6        9
##  0.31129 -0.08133  0.29409 -0.26995  0.01507 -0.25293 -0.01623
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.6944397  0.3009045  -2.308   0.0691 .
## Price        0.0056607  0.0004443  12.740   5.3e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2559 on 5 degrees of freedom
## Multiple R-squared:  0.9701, Adjusted R-squared:  0.9641
## F-statistic: 162.3 on 1 and 5 DF,  p-value: 5.299e-05

coef(linear_model)

##  (Intercept)        Price
## -0.694439675  0.005660744
```

```
y_pred<-predict(linear_model,data_test)
y_pred

##        2        5        7
## 1.230213 1.179266 1.434000

mse <- sum((y_pred-data_test$Discount)^2)
mse

## [1] 0.1053175

rmse <- mse^0.5
rmse

## [1] 0.3245266
```

#Create a Multivariate Linear Resgression model.

```
mat<-sample(seq(60,100,5),50,replace = TRUE)
che<-sample(seq(60,100,5),50,replace = TRUE)
phy<-sample(seq(60,100,5),50,replace = TRUE)
rv <- sample(c(2.5,-3.8),50,replace=TRUE)
cutoff<-mat+che*0.5+phy*0.5
cutoff<-cutoff+rv
df2<-data.frame(Maths=mat,Chemistry=che,Physics=phy,Cutoff=cutoff)
head(df2)

##   Maths Chemistry Physics Cutoff
## 1    70        95      70  155.0
## 2    60        60      75  123.7
## 3    80        65      70  143.7
## 4    95        75      85  171.2
## 5    60       100      90  151.2
## 6    75       100      60  157.5

ind=sample(1:50,35)
data_train <- df2[ind,]
data_test <- df2[-ind,]
linear_model <- lm(Cutoff ~ Maths+Chemistry+Physics,data=data_train)
summary(linear_model)

##
## Call:
## lm(formula = Cutoff ~ Maths + Chemistry + Physics, data = data_train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.763 -3.552  1.732  2.414  3.291
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.76790    5.58728  -0.674    0.505
```

```
## Maths         1.00237    0.04671  21.460  < 2e-16 ***
## Chemistry     0.53897    0.03996  13.487 1.64e-14 ***
## Physics       0.50768    0.04323  11.743 6.05e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.184 on 31 degrees of freedom
## Multiple R-squared:  0.9665, Adjusted R-squared:  0.9632
## F-statistic: 297.7 on 3 and 31 DF,  p-value: < 2.2e-16

y_pred<-predict(linear_model,data_test)
y_pred

##        1        3        8       10       15       18       24       29
## 153.1380 146.9925 167.3913 154.4778 137.3468 154.9857 170.8034 193.3627
##       32       37       38       39       41       44       45
## 144.2977 180.3313 190.7329 144.9620 145.2749 152.7337 149.4659

mse <- sum((y_pred-data_test$Cutoff)^2)
mse

## [1] 143.3228

rmse<-mse^0.5
rmse

## [1] 11.97175

ggplot(data_test,aes(Maths))+geom_point(aes(y=Cutoff))+geom_line(aes(y=y_pred
))
```



Common Question

#height (cms) vs age (years) for 120 persons by taking uniformly distributed values for age [10:70] and compute height using h = a * R + 76; where R is the random value in (6, 6.5, 7)

#1. Create the synthetic data.

```
age <- sample(10:70,120,replace=TRUE)
rv <- sample(c(6, 6.5, 7),120,replace=TRUE)
height <- 0.2*age*rv+76
df1<-data.frame(Age=age,Height=height)
head(df1)
```

```
##   Age Height
## 1  62  162.8
## 2  57  155.8
## 3  22  106.8
## 4  22  106.8
## 5  38  125.4
## 6  61  155.3
```

#2. Use simple linear regression and find the model.

```
ind=sample(1:120,84)
data_train <- df1[ind,]
data_test <- df1[-ind,]
linear_model <- lm(Height~Age,data=data_train)
summary(linear_model)
```

```
##
## Call:
## lm(formula = Height ~ Age, data = data_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.5514 -2.9084 -0.1928  3.0494  6.2697
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 75.81313    1.15395   65.70   <2e-16 ***
## Age          1.31055    0.02584   50.71   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.743 on 82 degrees of freedom
## Multiple R-squared:  0.9691, Adjusted R-squared:  0.9687
## F-statistic:  2572 on 1 and 82 DF,  p-value: < 2.2e-16
```

```
coef(linear_model)
```

```
## (Intercept)         Age
##   75.813133    1.310546
```

```
y_pred<-predict(linear_model,data_test)
y_pred[1:5]
```

```
##        5        7       13       19       20
## 125.6139 100.7135 164.9303 119.0612 104.6452
```

#3. Do performance evaluation for the model.

```
mse <- sum((y_pred-data_test$Height)^2)
mse
```

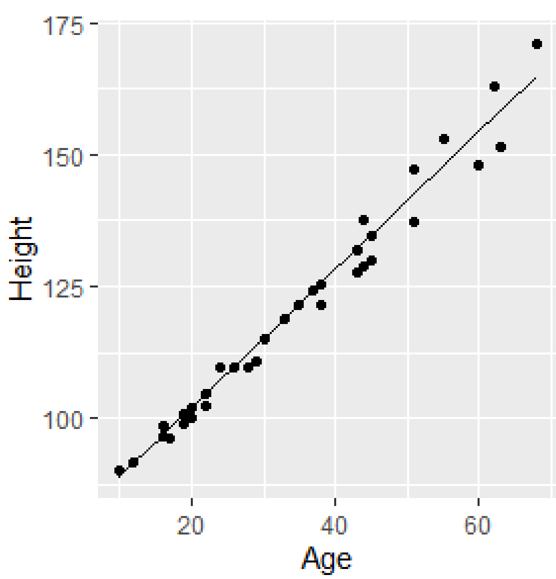
```
## [1] 388.0306
```

```
rmse <- mse^0.5
rmse
```

```
## [1] 19.69849
```

#4. Visualize the actual vs. predicted values.

```
ggplot(data_test,aes(Age))+geom_point(aes(y=Height))+geom_line(aes(y=y_pred))
```



#Logistic regression for the given California_housingdataset

```
library(MASS)
```

```
data <- housing
head(data)
```

```
##      Sat   Infl  Type Cont Freq
## 1    Low    Low Tower  Low   21
## 2 Medium    Low Tower  Low   21
```

```
## 3   High    Low Tower  Low   28


data$Freq<-ifelse(data$Freq>25,1,0)
head(data)

##      Sat   Infl  Type Cont Freq
## 1    Low    Low Tower  Low    0
## 2 Medium    Low Tower  Low    0
## 3   High    Low Tower  Low    1


ind=sample(1:72,50)
data_train <- data[ind,]
data_test <- data[-ind,]
log_model <- glm(Freq~.,data=data_train,family = binomial)
summary(log_model)
## Call:
## glm(formula = Freq ~ ., family = binomial, data = data_train)
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)        0.6628     1.4001   0.473   0.6359
## Sat.L             -0.7408     0.9397  -0.788   0.4305
## Sat.Q              2.4179     1.2176   1.986   0.0471 *
## InflMedium         1.1493     1.4116   0.814   0.4155
## InflHigh          -2.9915     1.5024  -1.991   0.0465 *
## TypeApartment      0.6535     1.2968   0.504   0.6143
## TypeAtrium       -22.9779  3656.8877  -0.006   0.9950
## TypeTerrace       -4.1870     1.8379  -2.278   0.0227 *
## ContHigh           1.6669     1.2014   1.387   0.1653
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 62.687  on 49  degrees of freedom
## Residual deviance: 24.886  on 41  degrees of freedom
## AIC: 42.886
##
## Number of Fisher Scoring iterations: 19

y_pred <- predict(log_model,data_test, type = "response")
y_pred

##            1            2           11           15           17
18
## 8.978659e-01 2.122471e-01 3.412058e-01 9.492561e-01 2.534611e-02
2.293563e-01
##           21           24           32           36           37
38
## 3.235094e-10 1.020995e-09 1.275366e-02 2.346619e-03 9.789707e-01
```

```
y_pred <- ifelse(y_pred >0.5, 1, 0)
y_pred

##  1  2 11 15 17 18 21 24 32 36 37 38 40 41 42 43 46 49 55 69 70 71
##  1  0  0  1  0  0  0  0  0  0  1  1  1  1  1  1  1  1  0  0  0  0

mean(y_pred == data_test$Freq)

## [1] 0.6818182
```

#Program to predict diabetes Pima Indian Diabetes dataset using Logistic Regression Classifier.

```
data <- Pima.tr2
data<-data[rowSums(is.na(data)) == 0, ]
head(data)

##   npreg glu bp skin  bmi   ped age type
## 1     5  86 68   28 30.2 0.364  24   No
## 2     7 195 70   33 25.1 0.163  55  Yes
## 3     5  77 82   41 35.8 0.156  35   No


ind=sample(1:200,160)
data_train <- data[ind,]
data_test <- data[-ind,]
log_model <- glm(type~.,data=data_train,family = binomial)
summary(log_model)
## Call:
## glm(formula = type ~ ., family = binomial, data = data_train)
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -10.277630   1.999342  -5.141 2.74e-07 ***
## npreg         0.159694   0.071526   2.233  0.02557 *
## glu           0.033160   0.007599   4.364 1.28e-05 ***
## bp            0.003767   0.020921   0.180  0.85713
## skin         -0.011889   0.026353  -0.451  0.65187
## bmi           0.085330   0.048203   1.770  0.07669 .
## ped           2.116938   0.748305   2.829  0.00467 **
## age           0.028674   0.024987   1.148  0.25116
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 205.92  on 159  degrees of freedom
## Residual deviance: 140.18  on 152  degrees of freedom
## AIC: 156.18
##
## Number of Fisher Scoring iterations: 5
```

```
y_pred <- predict(log_model,data_test, type = "response")
y_pred <- ifelse(y_pred >0.5, "Yes", "No")
y_pred

##     3     6    11    16    20    24    25    32    33    36    38    42
45
##  "No"  "No" "Yes"  "No"  "No"  "No"  "No"  "No"  "No" "Yes"  "No"  "No"
"No"
##    53    65    66    74    80    89    99   103   105   106   108   119
120
## "Yes"  "No"  "No"  "No" "Yes"  "No"  "No"  "No"  "No"  "No"  "No"  "No"
"Yes"

mean(y_pred == data_test$type)

## [1] 0.75
```

#Program to predict whether the client will subscribe (1/0) to a term deposit (variable y).

```
library(readr)
data <- read_delim("C:/Users/DELL/Downloads/bank-additional/bank-
additional/bank-additional.csv",delim = ";")

data<-data[rowSums(is.na(data)) == 0, ]
data$y<-as.factor(data$y)
head(data)

## # A tibble: 6 × 21
##     age job    marital education default housing loan  contact month
day_of_week
##   <dbl> <chr>  <chr>   <chr>     <chr>   <chr>   <chr> <chr>   <chr> <chr>
## 1    30 blue-… married basic.9y  no      yes     no    cellul… may   fri
## 2    39 servi… single  high.sch… no      no      no    teleph… may   fri
## 3    25 servi… married high.sch… no      yes     no    teleph… jun   wed
## 4    38 servi… married basic.9y  no      unknown unkn… teleph… jun   fri
## 5    47 admin. married universi… no      yes     no    cellul… nov   mon
## 6    32 servi… single  universi… no      no      no    cellul… sep   thu
## # ℹ 11 more variables: duration <dbl>, campaign <dbl>, pdays <dbl>,
## #   previous <dbl>, poutcome <chr>, emp.var.rate <dbl>, cons.price.idx
<dbl>,
## #   cons.conf.idx <dbl>, euribor3m <dbl>, nr.employed <dbl>, y <fct>

ind=sample(1:4119,2883)
data_train <- data[ind,]
data_test <- data[-ind,]
log_model <- glm(y~.,data=data_train,family = binomial)
summary(log_model)

##
## Call:
## glm(formula = y ~ ., family = binomial(link = "logit"), data = data_train)
##
```

```
## Coefficients: (1 not defined because of singularities)
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -2.570e+02  1.574e+02  -1.633 0.102507
## age                           4.560e-03  1.031e-02   0.442 0.658223
## jobstudent                    1.262e-01  5.208e-01   0.242 0.808565
## jobtechnician                 1.422e-01  2.893e-01   0.491 0.623175
## jobunemployed                 5.369e-01  4.951e-01   1.084 0.278173
## jobunknown                   -3.914e-01  9.365e-01  -0.418 0.675981
## maritalmarried                2.003e-01  3.071e-01   0.652 0.514302
## maritalsingle                 1.946e-02  3.586e-01   0.054 0.956728
## educationprofessional.course -1.874e-01  4.165e-01  -0.450 0.652841
## defaultyes                   -8.515e+00  5.354e+02  -0.016 0.987312
## housingunknown               -5.590e-01  6.228e-01  -0.898 0.369446
## housingyes                   -1.466e-01  1.732e-01  -0.846 0.397346
## loanunknown                          NA         NA      NA       NA
## loanyes                      -3.979e-02  2.346e-01  -0.170 0.865286
## contacttelephone             -1.358e+00  3.823e-01  -3.552 0.000382 ***
## monthaug                      9.784e-01  5.260e-01   1.860 0.062905 .
## monthdec                      1.823e+00  8.706e-01   2.094 0.036290 *
## monthjul                      9.695e-02  4.607e-01   0.210 0.833330
## monthsep                      6.583e-01  7.452e-01   0.883 0.377014
## day_of_weekmon               -3.749e-02  2.698e-01  -0.139 0.889486
## duration                      5.942e-03  3.408e-04  17.434  < 2e-16 ***
## campaign                     -1.382e-01  5.885e-02  -2.349 0.018840 *
## pdays                        -4.119e-04  8.303e-04  -0.496 0.619858
## previous                      7.962e-02  2.274e-01   0.350 0.726209
## poutcomenonexistent           7.199e-01  3.910e-01   1.841 0.065585 .
## poutcomesuccess               1.785e+00  8.235e-01   2.168 0.030160 *
## emp.var.rate                 -1.139e+00  5.996e-01  -1.899 0.057550 .
## cons.price.idx                2.168e+00  1.047e+00   2.070 0.038463 *
## cons.conf.idx                 7.342e-02  3.330e-02   2.204 0.027492 *
## euribor3m                    -4.004e-01  5.225e-01  -0.766 0.443504
## nr.employed                   1.038e-02  1.258e-02   0.825 0.409408
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##     Null deviance: 2005.8  on 2882  degrees of freedom
## Residual deviance: 1034.8  on 2830  degrees of freedom
## AIC: 1140.8
##
## Number of Fisher Scoring iterations: 12
```

```
y_pred <- predict(log_model,data_test, type = "response")
y_pred <- ifelse(y_pred >0.5, "Yes", "No")
y_pred[1:10]
```

```
##     1     2     3     4     5     6     7     8     9    10
##  "No"  "No"  "No" "Yes"  "No"  "No"  "No"  "No"  "No"  "No"
```