loading the library to read the CSV file

library(readxl)

Setting the path to the folder

setwd("C:/Users/Cholpon/Downloads")

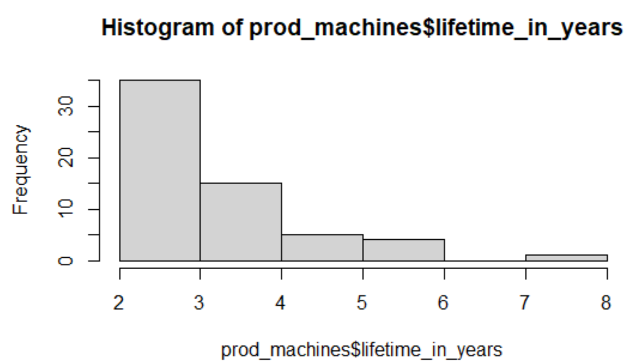
Loading the dataset

prod_machines <- read.csv('production machines.csv')

question 1a

plotting a histogram using a basic R function. The distribution is exponential

hist(prod_machines$lifetime_in_years)

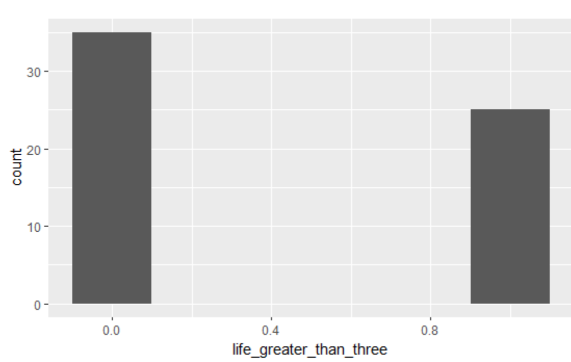


plotting a frequency histogram using a ggplot function

library(ggplot2)

ggplot(data=prod_machines,aes(life_greater_than_three))+geom_histogram(binwidth = 0.2)

this frequency histogram shows that value 0 appeared 35 times and 1 appeared 25 times

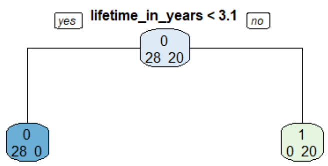


question 2

```
# random sampling
# step 1
training_index <- sample(1:nrow(prod_machines), size = 0.8*nrow(prod_machines))
# step 2
my_train <- prod_machines[training_index,]
my_test  <- prod_machines[-training_index,]
# looking at correlation and causation
cor.test(prod_machines$life_greater_than_three, prod_machines$production_units_lifetime)
cor.test(prod_machines$life_greater_than_three, prod_machines$lifetime_in_years)
# after looking into correlation of 'life_greater_than_three' variable with all the
# others, only two will be used for the regression and tree - production_units_lifetime
# with 0.6455844 correclation and lifetime_in_years with 0.8273851 correlation
# building a logistic regression
my_logit <- glm(life_greater_than_three ~ production_units_lifetime +
                  lifetime_in_years,
                data = my_train, family = "binomial")
summary(my_logit)
#console
> summary(my_logit)

Call:
 glm(formula = life_greater_than_three ~ production_units_lifetime +
     lifetime_in_years, family = "binomial", data = my_train)

Deviance Residuals:
 Min        1Q     Median       3Q       Max
-1.226e-04  -2.100e-08  -2.100e-08   2.100e-08   1.454e-04

Coefficients:
```

```
  Estimate Std. Error z value Pr(>|z|)
(Intercept)              -1.159e+03  2.547e+05  -0.005    0.996
production_units_lifetime -5.576e-03  3.371e+01   0.000    1.000
lifetime_in_years         3.809e+02  8.486e+04   0.004    0.996

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 6.5203e+01  on 47  degrees of freedom
Residual deviance: 4.4076e-08  on 45  degrees of freedom
AIC: 6

Number of Fisher Scoring iterations: 25

# The coefficients of the logistic regression give following insights:
# with every unit increase in 'lifetim e_in_years' the variable 'life_greater_than_three'
# increases by 381 units. And 'production_units_lifetime' had a negative
# impact.

# designing a Gini tree
# loading the library
library(rpart)
library(rpart.plot)

# building a tree on a train data using same variables as for the logistic regression.
my_tree <- rpart(life_greater_than_three ~ production_units_lifetime +
                   lifetime_in_years,
                 data = my_train, method = "class",
                 cp = 0.01)
```

rpart.plot(my_tree, type = 1, extra = 1)



The Gini Tree demonstrates that the values of 'lifetime_in_years' variables that are less than 3.1 will results in business success, meaning that 'life_greater_than_tree' variable will be 1.

building our confusion matrix

loading the library

library(caret)

using the PREDICT function on the test data

my_prediction <- predict(my_logit, my_test, type = 'response')

building the confusion matrix on the test data

my_conf <- confusionMatrix(data = as.factor(as.numeric(my_prediction > 0.5)),

reference = as.factor(as.numeric(my_test$life_greater_than_three)))

my_conf$table

concole

my_conf$table

Reference

Prediction 0 1

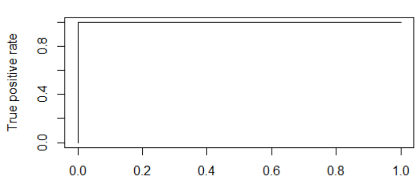
0 7 0

1 0 5

confusion matrix on training

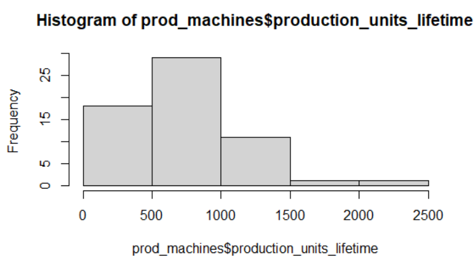
my_prediction_train <- predict(my_logit, my_train, type = 'response')

```
my_conf_train <- confusionMatrix(data = as.factor(as.numeric(my_prediction_train > 0.5)),
            reference = as.factor(as.numeric(my_train$life_greater_than_three)))

my_conf_train$table
# colsole
>my_conf_train$table
Reference
Prediction  0  1
       0 28  0
       1  0 20

# creating lifts and gains for logistic
# loading the library
library(ROCR)
# using the PREDICTION function
pred_val_logit <- prediction(my_prediction, my_test$life_greater_than_three)
# using the PERFORMANCE function
perf_logit <- performance(pred_val_logit, "tpr", "fpr")
# visualising the results
plot(perf_logit)
```

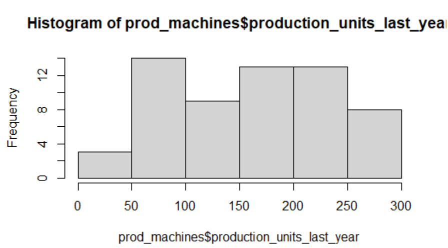



```
# comparing this gini with logit performance
my_tree_predict <- predict(my_tree, my_test,
                    type = "prob")
```

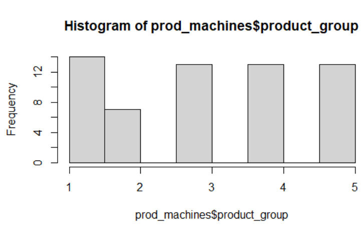
```
my_tree_prediction <- prediction(my_tree_predict[, 2], my_test$life_greater_than_three)

my_tree_perf <- performance(my_tree_prediction, "tpr", "fpr")

plot(perf_logit, col = "blue")
plot(my_tree_perf, col = "green", add = TRUE)

# question 3
# describing the distribution of variables
hist(prod_machines$production_units_lifetime) # the histogram shows exponential distribution
```



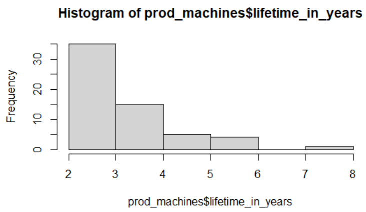

```
hist(prod_machines$production_units_last_year) # the histogram shows Gaussian distribution
```



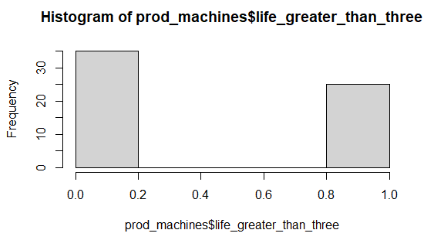

```
hist(prod_machines$product_group) # the histogram shows uniform distribution with exception at 2
```




```
hist(prod_machines$lifetime_in_years) # the histogram shows Gaussian distribution, with the values exponentially decreasing
```



hist(prod_machines$life_greater_than_three) # the histogram shows binomial distribution




```
# function to calculate mean and standard deviation
mean_std_func <- function(x = data.frame(), col_idx){
  my_mean <- mean(x[,col_idx])
  my_std <- sd(x[,col_idx])
  return (c(my_mean, my_std))
} #closing the function


# production_units_lifetime
mean_std_func(prod_machines, 2)
[1] 684.5367 395.7410 # there are some outliers in the distribution increasing the mean and standard deviation
# production_units_last_year
mean_std_func(prod_machines, 3)
[1] 159.53333  74.07291 # based on the histogram, the mean and st dev are normal, no big outliers
# product_group
mean_std_func(prod_machines, 4)
```

[1] 3.066667 1.471384 # this is a uniform distribution, the standard deviation is created because of the value 2

lifetime_in_years

mean_std_func(prod_machines, 5)# due to the nature of the distribution, it is not logical to draw the conclusion of the mean and std deviation as the values decrease exponentially.

[1] 3.158333 1.410889

life_greater_than_three

mean_std_func(prod_machines, 6) # binomial distribution with values of 1 and 0, which gives the mean and std of 0.41 and 0.49. When distribution is equal the mean should be 0.5, however in present case distribution is 35 in 0 and 25 in 1, which gives the mean of 0.41

[1] 0.4166667 0.4971671