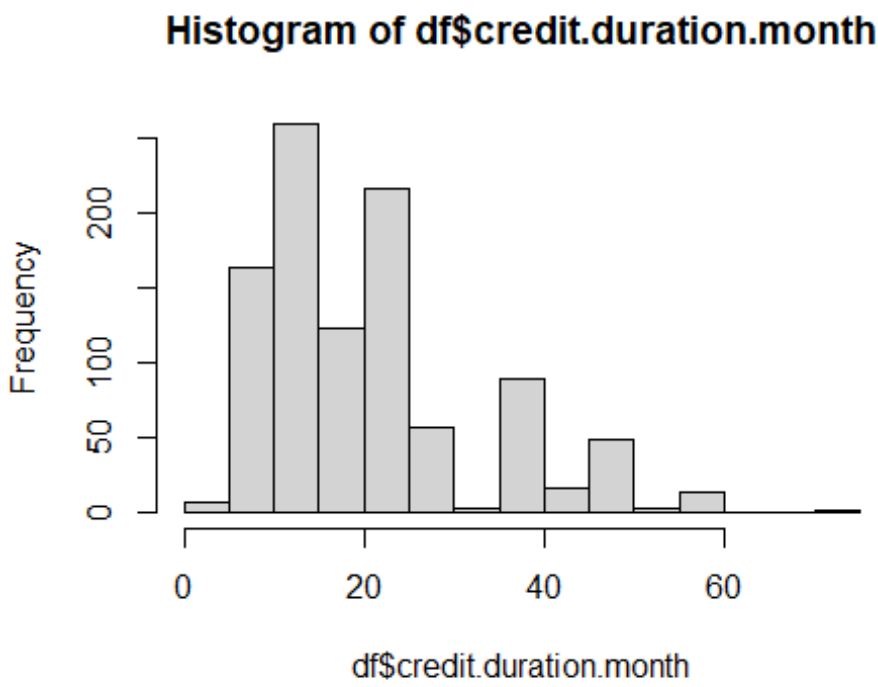
Assign4-1_Small

R Markdown

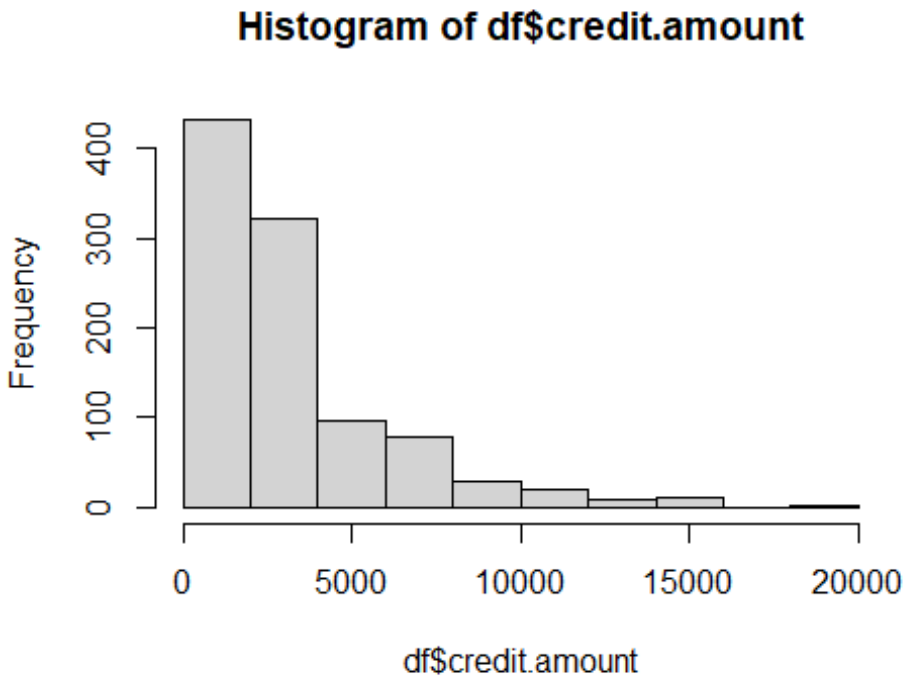
```
library(caret)
library(tidyverse)
library(ggplot2)
library(magrittr)
library(BBmisc)
library(e1071)
library(caret)

df <- read.csv("/Users/Nick/Desktop/credit2.csv", stringsAsFactors=FALSE)
str(df)

## 'data.frame':    1000 obs. of  21 variables:
##  $ credit.rating                 : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ account.balance               : int  1 1 2 1 1 1 1 1 3 2 ...
##  $ credit.duration.months        : int  18 9 12 12 12 10 8 6 18 24 ...
##  $ previous.credit.payment.status: int  3 3 2 3 3 3 3 3 3 2 ...
##  $ credit.purpose                : int  2 4 4 4 4 4 4 4 3 3 ...
##  $ credit.amount                 : int  1049 2799 841 2122 2171 2241 3398
1361 1098 3758 ...
##  $ savings                       : int  1 1 2 1 1 1 1 1 1 3 ...
##  $ employment.duration           : int  1 2 3 2 2 1 3 1 1 1 ...
##  $ installment.rate              : int  4 2 2 3 4 1 1 2 4 1 ...
##  $ marital.status                : int  1 3 1 3 3 3 3 3 1 1 ...
##  $ guarantor                     : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ residence.duration            : int  4 2 4 2 4 3 4 4 4 4 ...
##  $ current.assets                : int  2 1 1 1 2 1 1 1 3 4 ...
##  $ age                           : int  21 36 23 39 38 48 39 40 65 23 ...
##  $ other.credits                 : int  2 2 2 2 1 2 2 2 2 2 ...
##  $ apartment.type                : int  1 1 1 1 2 1 2 2 2 1 ...
##  $ bank.credits                  : int  1 2 1 2 2 2 2 1 2 1 ...
##  $ occupation                    : int  3 3 2 2 2 2 2 2 1 1 ...
##  $ dependents                    : int  1 2 1 2 1 2 1 2 1 1 ...
##  $ telephone                     : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ foreign.worker                : int  1 1 1 2 2 2 2 2 1 1 ...
```

The below demonstrates the three Histograms for Credit Duration Month, Credit Amount, and Age.

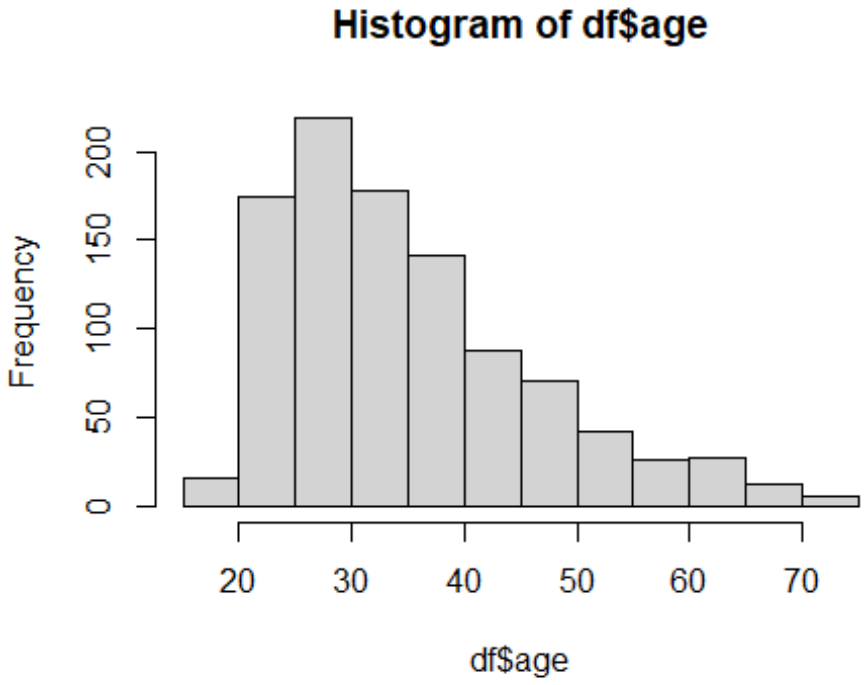
```
hist(df$credit.duration.month)
```




```
hist(df$credit.amount)
```


Histogram of df$credit.amount
Frequency
0
100
200
300
400
0
5000
10000
15000
20000
df$credit.amount

```r
hist(df$age)
```


Histogram of df$age
Frequency
0
50
100
150
200
20
30
40
50
60
70
df$age

```
numeric.var <- c("account.balance", "credit.rating", "previous.credit.payment
.status", "credit.purpose", "savings", "employment.duration", "installment.ra
te", "marital.status", "guarantor", "residence.duration", "current.assets", "
other.credits", "apartment.type", "bank.credits", "occupation", "dependents",
"telephone", "foreign.worker")
```

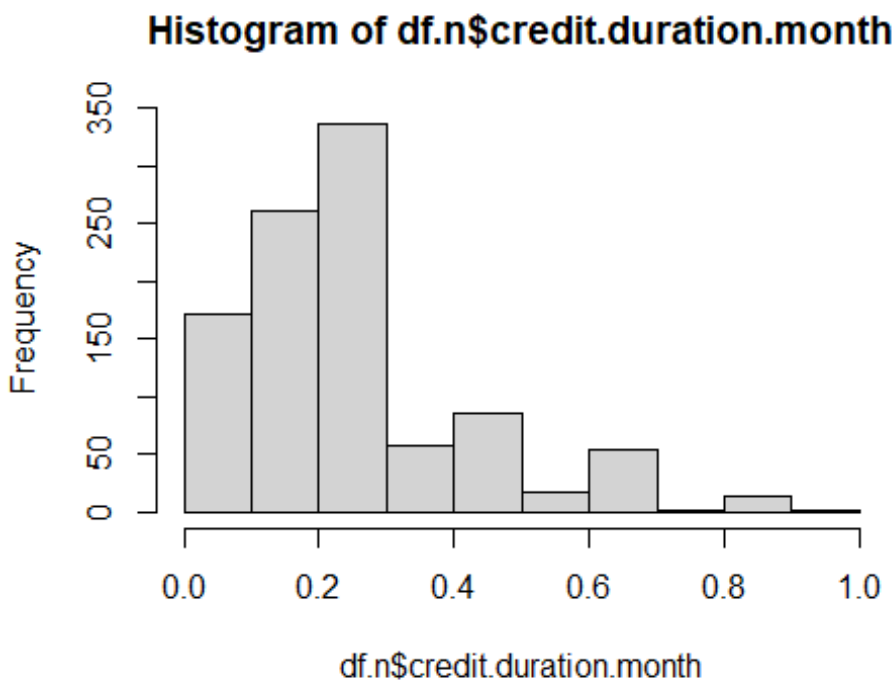
Normalization of the Data

```
normalize <- function(x) {return ((x - min(x)) / (max(x) - min(x)))}
df.n1 <- as.data.frame(lapply(df[3], normalize))
df.n2 <- as.data.frame(lapply(df[6], normalize))
df.n3 <- as.data.frame(lapply(df[14], normalize))

df.n <- cbind(df.n1, df.n2, df.n3)
df.n <- cbind(df,df.n)
df.n <- subset(df.n, , -c(3,6,14))
str(df.n)
```

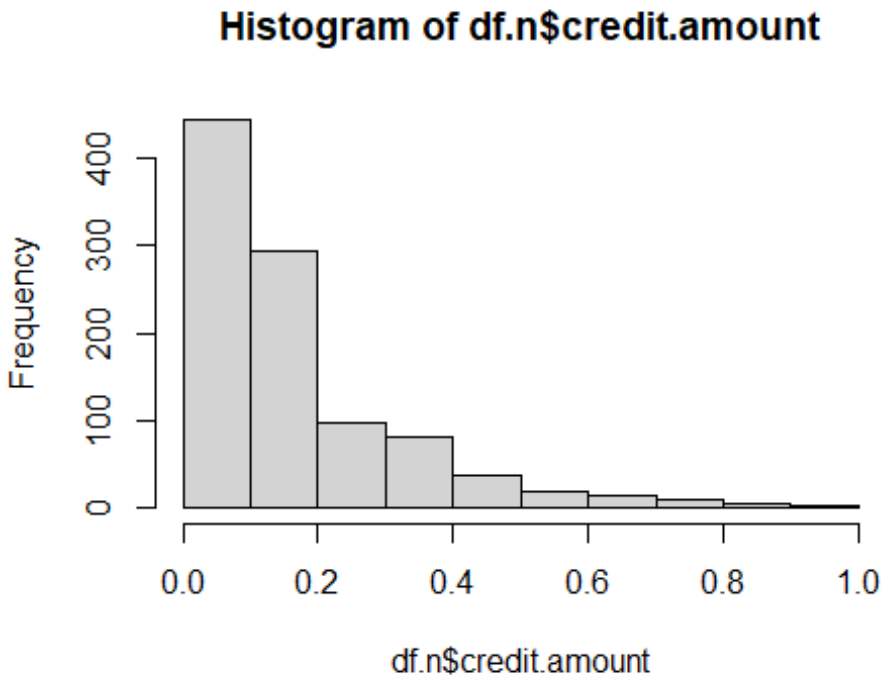
```
## 'data.frame':    1000 obs. of  21 variables:
##  $ credit.rating                 : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ account.balance               : int  1 1 2 1 1 1 1 1 3 2 ...
##  $ previous.credit.payment.status: int  3 3 2 3 3 3 3 3 3 2 ...
##  $ credit.purpose                : int  2 4 4 4 4 4 4 4 3 3 ...
##  $ savings                       : int  1 1 2 1 1 1 1 1 1 3 ...
##  $ employment.duration           : int  1 2 3 2 2 1 3 1 1 1 ...
##  $ installment.rate              : int  4 2 2 3 4 1 1 2 4 1 ...
##  $ marital.status                : int  1 3 1 3 3 3 3 3 1 1 ...
##  $ guarantor                     : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ residence.duration            : int  4 2 4 2 4 3 4 4 4 4 ...
##  $ current.assets                : int  2 1 1 1 2 1 1 1 3 4 ...
##  $ other.credits                 : int  2 2 2 2 1 2 2 2 2 2 ...
##  $ apartment.type                : int  1 1 1 1 2 1 2 2 2 1 ...
##  $ bank.credits                  : int  1 2 1 2 2 2 2 1 2 1 ...
##  $ occupation                    : int  3 3 2 2 2 2 2 2 1 1 ...
##  $ dependents                    : int  1 2 1 2 1 2 1 2 1 1 ...
##  $ telephone                     : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ foreign.worker                : int  1 1 1 2 2 2 2 2 1 1 ...
##  $ credit.duration.months        : num  0.2059 0.0735 0.1176 0.1176 0.1176
...
##  $ credit.amount                 : num  0.044 0.1403 0.0325 0.103 0.1057 .
..
##  $ age                           : num  0.0357 0.3036 0.0714 0.3571 0.3393
...
```

The three historgrams below are the normalized version of the same ones above . There is a slight difference using normalization.

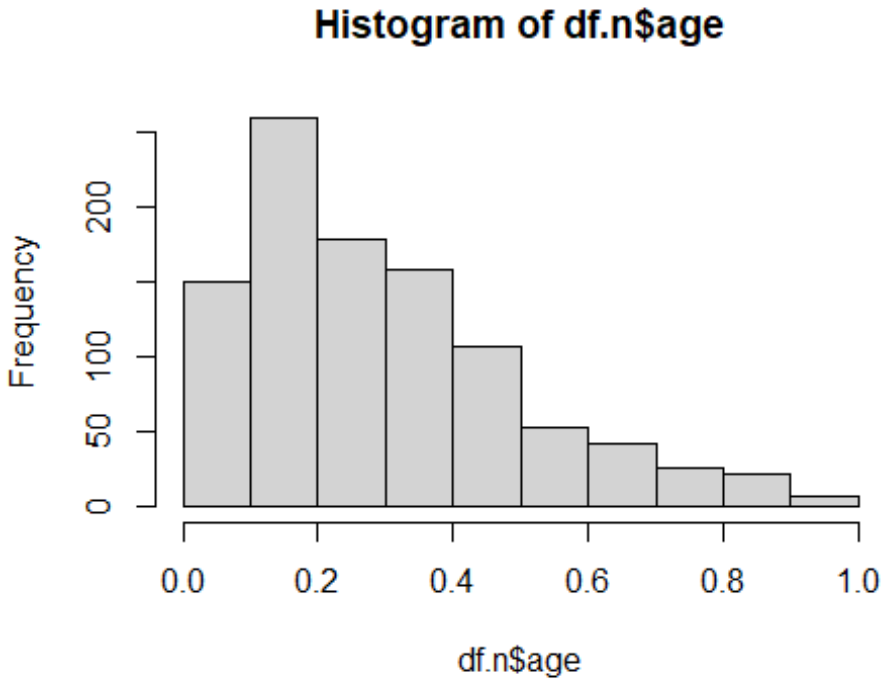
```
hist(df.n$credit.duration.month)
```




```
hist(df.n$credit.amount)
```




```r
hist(df.n$age)
```



Training the Data set

```
set.seed(1234)
intrain <- createDataPartition(y=df.n$credit.rating, p=0.6, list=FALSE)
train <- df.n[intrain,]
test <- df.n[-intrain,]
```

Support Vector Machine

```
#Default
credit.svm = svm(credit.rating~ ., data = train, cost = 1, gamma = 1/length(t
rain), probability = TRUE)

#Linear
credit.svm2 = svm(credit.rating~ ., data = train, kernel = "linear", cost = 1
, gamma = 1/length(train), probability = TRUE)

#Non-Linear
credit.svm3 = svm(credit.rating~ ., data = train, kernel = "radial", cost = 1
, gamma = 1/length(train), probability = TRUE)

#Default
prob.svm = predict(credit.svm, train, probability = TRUE)
pred.svm = as.numeric((prob.svm >= 0.16))

table(train$credit.rating, pred.svm, dnn = c("Obs", "Pred"))
```

```
##    Pred
## Obs   0   1
##   0  34 147
##   1   0 419
```

```
agreement <- pred.svm==test$credit.rating

table(agreement)
```

```
## agreement
## FALSE  TRUE
##   156   444
```

```
prop.table(table(agreement))
```

```
## agreement
## FALSE  TRUE
##  0.26  0.74
```

```
prob.svm1 = predict(credit.svm, test, probability = TRUE)
pred.svm1 = as.numeric((prob.svm1 >= 0.16))

table(test$credit.rating, pred.svm1, dnn = c("Obs", "Pred"))
```

```
##    Pred
## Obs   0   1
##   0   5 114
##   1   0 281
```

```
agreement1 <- pred.svm1==test$credit.rating
table(agreement1)
```

```
## agreement1
## FALSE  TRUE
##   114   286
```

```
prop.table(table(agreement1))
```

```
## agreement1
## FALSE  TRUE
## 0.285 0.715
```

```
#Linear
prob.svm2 = predict(credit.svm2, train, probability = TRUE)
pred.svm2 = as.numeric((prob.svm2 >= 0.16))

table(train$credit.rating, pred.svm2, dnn = c("Obs", "Pred"))
```

```
##    Pred
## Obs   0   1
##   0   3 178
##   1   0 419
```

```
agreement2 <- pred.svm2==test$credit.rating
```

```
## Warning in pred.svm2 == test$credit.rating: longer object length is not a
## multiple of shorter object length
```

```
table(agreement2)
```

```
## agreement2
## FALSE  TRUE
##   127   473
```

```
prop.table(table(agreement2))
```

```
## agreement2
##     FALSE      TRUE
## 0.2116667 0.7883333
```

```
prob.svm3 = predict(credit.svm2, test, probability = TRUE)
pred.svm3 = as.numeric((prob.svm3 >= 0.16))

table(test$credit.rating, pred.svm3, dnn = c("Obs", "Pred"))
```

```
##    Pred
## Obs   0   1
```

```
##   0   1 118
##   1   0 281
```

```
agreement3 <- pred.svm3==test$credit.rating
table(agreement3)
```

```
## agreement3
## FALSE  TRUE
##   118   282
```

```
prop.table(table(agreement3))
```

```
## agreement3
## FALSE  TRUE
## 0.295 0.705
```

```
#Non-Linear
prob.svm4 = predict(credit.svm3, train, probability = TRUE)
pred.svm4 = as.numeric((prob.svm4 >= 0.16))

table(train$credit.rating, pred.svm4, dnn = c("Obs", "Pred"))
```

```
##    Pred
## Obs   0   1
##   0  34 147
##   1   0 419
```

```
agreement4 <- pred.svm==test$credit.rating
```

```
## Warning in pred.svm == test$credit.rating: longer object length is not a
## multiple of shorter object length
```

```
table(agreement4)
```

```
## agreement4
## FALSE  TRUE
##   156   444
```

```
prop.table(table(agreement4))
```

```
## agreement4
## FALSE  TRUE
##  0.26  0.74
```

```
prob.svm5 = predict(credit.svm3, test, probability = TRUE)
pred.svm5 = as.numeric((prob.svm5 >= 0.16))

table(test$credit.rating, pred.svm5, dnn = c("Obs", "Pred"))
```

```
##    Pred
## Obs   0   1
##   0   5 114
##   1   0 281
```

```r
agreement5 <- pred.svm==test$credit.rating

## Warning in pred.svm == test$credit.rating: longer object length is not a
## multiple of shorter object length

table(agreement5)

## agreement5
## FALSE  TRUE
##   156   444

prop.table(table(agreement5))

## agreement5
## FALSE  TRUE
##  0.26  0.74
```

Conclusion: Comparing Default, Linear and Non-Linear SVm model we discover that the best model is the Linear model. The training predicted Linear model shows a 78.83% accuracy. In comparison to the Linear predicted test model, we can see it is 70.5% accurate which is very good. Non-Linear and Default models are relatively close in their accuracy but is less than the Linear predicted training model.