In this article, you learn how to make Automated data report storytelling in R for Credit Modelling. First you need to install the `rmarkdown` package into your R library. Assuming that you installed the `rmarkdown`, next you create a new `rmarkdown` script in R.

After this you type the following code in order to create a dashboard with rmarkdown and flexdashboard:

```{r setup, include=FALSE}

knitr::opts\_chunk$set(echo = FALSE)

library(flexdashboard)

# Load R packages into the library

# Data management packages

library(DescTools)

library(skimr)

library(plyr)

library(dplyr)

library(aod)

library(readxl)

# Visualization packages

library(Deducer)

library(ggplot2)

library(plotly)

# Machine learnning method packages

library(ROCR)

library(pROC)

library(caret)

library(MASS)

library(sjPlot)

# Import dataset

loan\_data <- read.csv("C:/Users/Bruger/Documents/R work/Credit modelling/loan.csv")

# Selecting the relevant variables in the dataset:

loan\_data <- loan\_data[,c("grade","sub\_grade","term","loan\_amnt","issue\_d","loan\_status","emp\_length",

"home\_ownership", "annual\_inc","verification\_status","purpose","dti",

"delinq\_2yrs","addr\_state","int\_rate", "inq\_last\_6mths","mths\_since\_last\_delinq",

"mths\_since\_last\_record","open\_acc","pub\_rec","revol\_bal","revol\_util","total\_acc")]

# Data management for missing observations

loan\_data$mths\_since\_last\_delinq[is.na(loan\_data$mths\_since\_last\_delinq)] <- 0

loan\_data$mths\_since\_last\_record[is.na(loan\_data$mths\_since\_last\_record)] <- 0

var.has.na <- lapply(loan\_data, function(x){any(is.na(x))})

num\_na <- which( var.has.na == TRUE )

per\_na <- num\_na/dim(loan\_data)[1]

loan\_data <- loan\_data[complete.cases(loan\_data),]

loan\_datahis<-loan\_data

# Focus on the historical loans

loan\_data=as.data.frame(loan\_data[loan\_data$loan\_status!="Current", ])

limits\_inc = quantile(loan\_data$annual\_inc, seq(0,1,0.1))

labels <- c(0, limits\_inc[2:10], "+inf")

labels <- prettyNum(labels, big.mark = ",")

labels <- paste(labels[1:10], labels[2:11], sep = "-")

loan\_data$annual\_inc <- cut(loan\_data$annual\_inc, limits\_inc, labels = labels, include.lowest = T)

loan\_data[,"annual\_inc"] <- as.character(loan\_data[,"annual\_inc"])

# Create binary variables for the logistic regression analysis

# Annual\_inc

loan\_data$annual\_inc[loan\_data$annual\_inc == "70,000- 80,000"| loan\_data$annual\_inc == "80,000- 94,000" | loan\_data$annual\_inc == "94,000-120,000" | loan\_data$annual\_inc == "120,000- +inf" ] <- 1

loan\_data$annual\_inc[loan\_data$annual\_inc != 1] <- 0

loan\_data$annual\_inc <- as.numeric(loan\_data$annual\_inc)

# Home\_ownership

loan\_data$home\_ownership <- as.character(loan\_data$home\_ownership)

loan\_data$home\_ownership[loan\_data$home\_ownership=="OWN" | loan\_data$home\_ownership=="MORTGAGE" ] <- 1

loan\_data$home\_ownership[loan\_data$home\_ownership!=1] <- 0

# Dealinq\_2yrs

loan\_data$delinq\_2yrs <- as.character(loan\_data$delinq\_2yrs)

loan\_data$delinq\_2yrs[loan\_data$delinq\_2yrs=="0"] <- 0

loan\_data$delinq\_2yrs[loan\_data$delinq\_2yrs!= 0] <- 1

# Verification status: if Verified = 1 ; otherwise = 0

loan\_data$verification\_status = as.character(loan\_data$verification\_status)

loan\_data$verification\_status[loan\_data$verification\_status == "Verified" | loan\_data$verification\_status == "Source Verified"] = 1

loan\_data$verification\_status[loan\_data$verification\_status != 1] = 0

loan\_data$verification\_status=as.numeric(loan\_data$verification\_status)

# Dti

dti\_quant <- quantile(loan\_data$dti, seq(0, 1, 0.1))

labels = c(0,prettyNum(dti\_quant[2:10], big.mark = ","), "+Inf")

labels = paste(labels[1:10],labels[2:11], sep = "-")

loan\_data <- mutate(loan\_data, dti= cut(loan\_data$dti, breaks = dti\_quant, labels = factor(labels), include.lowest = T))

loan\_data$dti <- as.character(loan\_data$dti)

loan\_data$dti[loan\_data$dti == "0-6.57" | loan\_data$dti == "12.13-14.32" | loan\_data$dti == "14.32-16.49" ] <- 1

loan\_data$dti[loan\_data$dti!=1] <- 0

# Status

loan\_data$loan\_status <- as.character(loan\_data$loan\_status)

loan\_data$loan\_status[loan\_data$loan\_status == "Charged Off" | loan\_data$loan\_status == "Default" ] <- 1

loan\_data$loan\_status[loan\_data$loan\_status != 1] <- 0

table(loan\_data$loan\_status)

PercTable(loan\_data$loan\_status)

# Change to nummeric variables:

loan\_data[,"revol\_util"] <- as.numeric(sub("%", "",loan\_data$"revol\_util", fixed =TRUE))/100

loan\_data[,"int\_rate"] <- as.numeric(sub("%", "",loan\_data$"int\_rate", fixed =TRUE))/100

loan\_data$loan\_status <- as.numeric(loan\_data$loan\_status)

# Grouping variables

loan\_data$purpose <- as.character(loan\_data$purpose)

loan\_data$purpose[loan\_data$purpose == "car" | loan\_data$purpose == "major\_purchase" |

loan\_data$purpose == "home\_improvement"| loan\_data$purpose == "credit\_card" ] <- 2

loan\_data$purpose[loan\_data$purpose == "moving" | loan\_data$purpose == "small\_business" |

loan\_data$purpose == "renewable\_energy" ] <- 0

loan\_data$purpose[loan\_data$purpose!= 0 & loan\_data$purpose!= 2 ] <- 1

loan\_data$purpose <- as.factor(loan\_data$purpose)

##Machine Learning: Multiple Logistic Regression Models

# Logistic: Logit stepwise Regression

logregmodI <- glm(loan\_status ~ loan\_amnt + home\_ownership + annual\_inc

+ verification\_status + purpose + dti + delinq\_2yrs

+ int\_rate + inq\_last\_6mths + mths\_since\_last\_delinq

+ revol\_bal + revol\_util + total\_acc,

data = loan\_data, family = binomial(link= "logit"))

step <- stepAIC(logregmodI, direction="both")

step$anova

# Create a training- and testing dataset

percing <- floor((nrow(loan\_data)/4)\*3)

loan <- loan\_data[sample(nrow(loan\_data)), ]

loan.training <- loan[1:percing, ]

loan.testing <- loan[(percing+1):nrow(loan), ]

# Begin training of the model

fitting.logistic <- glm(loan\_status ~ loan\_amnt + home\_ownership + verification\_status +

purpose + dti + delinq\_2yrs + int\_rate + inq\_last\_6mths +

mths\_since\_last\_delinq + revol\_bal + revol\_util + total\_acc,

data=loan.training,family = binomial(link= "logit"))

```

## R Markdown

This is an R Markdown presentation. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see .

### Chart A - Bar chart of the loan amount

The below output shows The loan amount with regards to number of loans

```{r, echo = FALSE}

# Visualization of the data

# Bar chart of the loan amount

loanamount\_barchart <- ggplot(data=loan\_data, aes(loan\_data$loan\_amnt)) +

geom\_histogram(breaks=seq(0, 35000, by=1000),

col="black", aes(fill=..count..)) +

scale\_fill\_gradient("Count", low="green1", high="yellowgreen")+

labs(title="Loan Amount", x="Amount", y="Number of Loans")

ggplotly(loanamount\_barchart)

```

### Chart B - Box plot of loan amount

The below plot shows a box plot of the loan amount with respect to different loan status

```{r, echo = FALSE}

# Box plot of loan amount

box\_plot\_stat <- ggplot(loan\_datahis, aes(loan\_status, loan\_amnt))

box\_plot\_stat + geom\_boxplot(aes(fill = loan\_status)) +

theme(axis.text.x = element\_blank()) +

labs(list(title = "Loan amount by status", x = "Loan Status", y = "Amount"))

```

### Chart C - Logistic regression

The below table is a logistic regression credit model. It displays the ods ratios in the regression model. An odds ratio is a relative measure of effect, which allows the comparison of a dichotom outcome. An odds ratio greater than 1 indicates that the condition or event is more likely to occur in the group. An odds ratio less than 1 indicates that the condition or event is less likely to occur in the group.

```{r}

tab\_model(fitting.logistic)

```

### Chart D - ROC graph visualizaiton

The below table is a ROC curve of the logistic regression credit model. The value displays how well the model have explained the outcome.

```{r}

# AUC and ROC curve

fitted.results <- predict(fitting.logistic, newdata = loan.testing, type = "response")

loan.testing$prob <- fitted.results

pred <- prediction(loan.testing$prob,loan.testing$loan\_status)

auc1 <- performance(pred, measure = "auc")

auc1@y.values

# Performance function

ROCRperf = performance(pred, "tpr", "fpr")

# Plot the ROC graph Add threshold labels

plot(ROCRperf, colorize=TRUE, print.cutoffs.at=seq(0,1,by=0.1), text.adj=c(-0.2,1.7))

abline(0, 1, col= "black")

```

Screenshot:  
[](https://i0.wp.com/datascienceplus.com/wp-content/uploads/2019/01/Data-report-story-telling.jpg?ssl=1)