R Notebook

Final EDA

Introduction

Credit Card Defaults can occur due to a number of reasons. According to the use case defintion > A default can occur when a borrower is unable to make timely payments, misses payments, or avoids/stops making payments

The question that we are concerend with is > Which priority clients have the highest risk of credit card default?

Libraries

```r
library(tidyverse)
```

```
## -- Attaching packages ------------------------------------------------------------- tidyverse 1.3.0 -
```

```
## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## -- Conflicts ---------------------------------------------------------------- tidyverse_conflicts() -
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(gmodels)
library(ggridges)
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```r
library(plotly)
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

Loading Data

```r
train_df <- read.csv("../credit_card_default_train.csv")
train_df$NEXT_MONTH_DEFAULT <- as.factor(train_df$NEXT_MONTH_DEFAULT)
test_df <- read.csv("../credit_card_default_test.csv")
print(nrow(train_df))
```

```
## [1] 24000
```

```r
print(colnames(train_df))
```

```
##  [1] "Client_ID"          "Balance_Limit_V1"   "Gender"
##  [4] "EDUCATION_STATUS"   "MARITAL_STATUS"     "AGE"
##  [7] "PAY_JULY"           "PAY_AUG"            "PAY_SEP"
## [10] "PAY_OCT"            "PAY_NOV"            "PAY_DEC"
## [13] "DUE_AMT_JULY"       "DUE_AMT_AUG"        "DUE_AMT_SEP"
## [16] "DUE_AMT_OCT"        "DUE_AMT_NOV"        "DUE_AMT_DEC"
## [19] "PAID_AMT_JULY"      "PAID_AMT_AUG"       "PAID_AMT_SEP"
## [22] "PAID_AMT_OCT"       "PAID_AMT_NOV"       "PAID_AMT_DEC"
## [25] "NEXT_MONTH_DEFAULT"
```

```r
print(str(train_df))
```

```
## 'data.frame':    24000 obs. of  25 variables:
##  $ Client_ID         : Factor w/ 24000 levels "A100","A1000",..: 8869 17782 18686 19584 20487 22231 2
##  $ Balance_Limit_V1  : Factor w/ 8 levels " 500K","1.5M",..: 4 4 3 6 4 8 3 3 1 4 ...
##  $ Gender            : Factor w/ 2 levels "F","M": 2 1 1 1 1 1 2 1 2 2 ...
##  $ EDUCATION_STATUS  : Factor w/ 3 levels "Graduate","High School",..: 1 2 2 1 1 1 3 2 3 3 ...
##  $ MARITAL_STATUS    : Factor w/ 2 levels "Other","Single": 1 1 2 2 1 2 2 1 1 2 ...
##  $ AGE               : Factor w/ 4 levels "31-45","46-65",..: 1 3 1 1 1 1 3 3 1 2 ...
##  $ PAY_JULY          : int  -1 0 4 2 2 0 1 2 0 0 ...
##  $ PAY_AUG           : int  -1 -1 3 0 2 0 2 2 0 0 ...
##  $ PAY_SEP           : int  -1 -1 2 0 0 0 2 2 0 2 ...
##  $ PAY_OCT           : int  -1 -1 2 0 0 0 2 0 2 0 ...
##  $ PAY_NOV           : int  -1 -1 -2 0 0 0 2 0 0 0 ...
##  $ PAY_DEC           : int  -1 0 -2 0 0 0 2 2 0 0 ...
##  $ DUE_AMT_JULY      : int  3248 353351 16681 90457 429556 361284 8991 51836 198579 268551 ...
##  $ DUE_AMT_AUG       : int  3389 151818 16082 92848 419466 364802 8515 55828 204634 282726 ...
##  $ DUE_AMT_SEP       : int  6004 26948 15477 95193 429785 366703 11698 54241 218092 274123 ...
##  $ DUE_AMT_OCT       : int  39418 43530 0 97309 435354 353910 11173 55325 212970 221148 ...
##  $ DUE_AMT_NOV       : int  162772 80811 0 100353 445271 356117 12030 59272 213654 222936 ...
##  $ DUE_AMT_DEC       : int  -13982 124590 0 102740 453899 358845 12647 57976 217992 224276 ...
##  $ PAID_AMT_JULY     : int  3437 151818 0 3855 0 16632 0 5521 9240 26565 ...
##  $ PAID_AMT_AUG      : int  6004 46200 0 3890 20790 18480 3696 0 17325 0 ...
##  $ PAID_AMT_SEP      : int  39418 43530 0 3696 16170 12728 0 1984 0 8184 ...
##  $ PAID_AMT_OCT      : int  162772 80811 0 4620 17325 13398 1386 4844 6930 8547 ...
##  $ PAID_AMT_NOV      : int  0 942 0 4049 16401 13860 1155 0 11550 8194 ...
##  $ PAID_AMT_DEC      : int  538165 33666 0 3918 17325 12705 0 2523 11550 7311 ...
##  $ NEXT_MONTH_DEFAULT: Factor w/ 2 levels "0","1": 1 1 2 2 1 1 1 2 1 1 ...
## NULL
```

```r
print(summary(train_df))
```

```
##    Client_ID     Balance_Limit_V1 Gender       EDUCATION_STATUS
##  A100    :     1   1M     :5951     F: 9540    Graduate   : 8478
##  A1000   :     1   200K   :5159     M:14460    High School: 3925
```

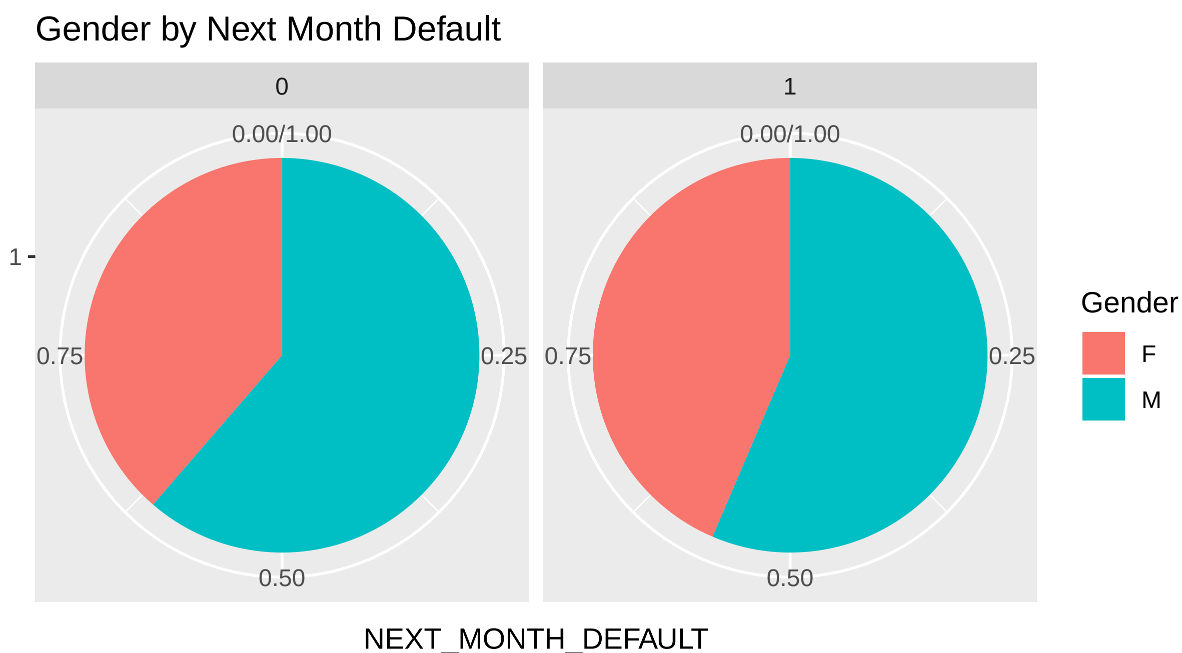
```
## A10000 :    1   100K   :3449              Other      :11597
## A10001 :    1   400K   :3065
## A10002 :    1    500K  :2790
## A10003 :    1   300K   :2411
## (Other):23994   (Other):1175
## MARITAL_STATUS           AGE           PAY_JULY           PAY_AUG
## Other :13070   31-45       :12124   Min.   :-2.00000   Min.   :-2.00
## Single:10930   46-65       : 4150   1st Qu.:-1.00000   1st Qu.:-1.00
##                Less than 30: 7638   Median : 0.00000   Median : 0.00
##                More than 65:   88   Mean   :-0.01421   Mean   :-0.13
##                                     3rd Qu.: 0.00000   3rd Qu.: 0.00
##                                     Max.   : 8.00000   Max.   : 8.00
##
##    PAY_SEP           PAY_OCT           PAY_NOV           PAY_DEC
## Min.   :-2.0000   Min.   :-2.0000   Min.   :-2.0000   Min.   :-2.0000
## 1st Qu.:-1.0000   1st Qu.:-1.0000   1st Qu.:-1.0000   1st Qu.:-1.0000
## Median : 0.0000   Median : 0.0000   Median : 0.0000   Median : 0.0000
## Mean   :-0.1587   Mean   :-0.2155   Mean   :-0.2612   Mean   :-0.2877
## 3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.0000
## Max.   : 8.0000   Max.   : 8.0000   Max.   : 8.0000   Max.   : 8.0000
##
##  DUE_AMT_JULY      DUE_AMT_AUG       DUE_AMT_SEP       DUE_AMT_OCT
## Min.   :-382490   Min.   :-161185   Min.   :-142079   Min.   :-392700
## 1st Qu.:   8246   1st Qu.:   6969   1st Qu.:   6238   1st Qu.:   5429
## Median :  51568   Median :  48717   Median :  46412   Median :  44105
## Mean   : 118870   Mean   : 114073   Mean   : 109244   Mean   : 100357
## 3rd Qu.: 156274   3rd Qu.: 148905   3rd Qu.: 140162   3rd Qu.: 126975
## Max.   :2228020   Max.   :2272881   Max.   :3844046   Max.   :2059564
##
##  DUE_AMT_NOV       DUE_AMT_DEC      PAID_AMT_JULY      PAID_AMT_AUG
## Min.   :-187882   Min.   :-784483   Min.   :      0   Min.   :      0
## 1st Qu.:   4180   1st Qu.:   2913   1st Qu.:   2310   1st Qu.:   1956
## Median :  41863   Median :  39409   Median :   4920   Median :   4646
## Mean   :  93777   Mean   :  90341   Mean   :  13306   Mean   :  13867
## 3rd Qu.: 116926   3rd Qu.: 114435   3rd Qu.:  11605   3rd Qu.:  11550
## Max.   :2141765   Max.   :2221444   Max.   :2017905   Max.   :3890638
##
##  PAID_AMT_SEP      PAID_AMT_OCT      PAID_AMT_NOV     PAID_AMT_DEC
## Min.   :      0   Min.   :      0   Min.   :     0   Min.   :      0
## 1st Qu.:    901   1st Qu.:    693   1st Qu.:   610   1st Qu.:    307
## Median :   4197   Median :   3465   Median :  3465   Median :   3465
## Mean   :  12093   Mean   :  11225   Mean   : 11175   Mean   :  12301
## 3rd Qu.:  10626   3rd Qu.:   9360   3rd Qu.:  9412   3rd Qu.:   9252
## Max.   :2069852   Max.   :1434510   Max.   :965557   Max.   :1221218
##
## NEXT_MONTH_DEFAULT
## 0:18670
## 1: 5330
##
##
##
##
##
```

Categorical Variables

First let's explore the Categorical variables. Since we are interested in what are the factors that affect the risk of credit card default, we shall be exploring considering the two levels of the response variable

Gender

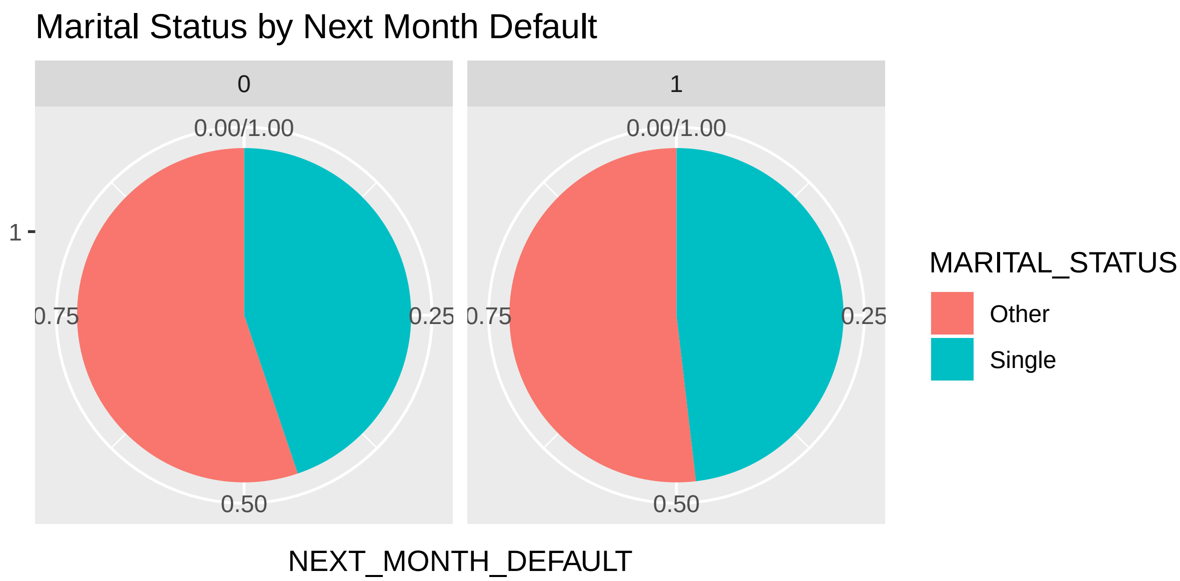
```
train_df$Gender <- factor(train_df$Gender) # converts to a categorical variable
train_df$NEXT_MONTH_DEFAULT <- factor(train_df$NEXT_MONTH_DEFAULT) # converts to a categorical variable
p1 <- ggplot(data=train_df, aes(x=factor(1), stat="bin", fill=Gender)) +
  geom_bar(position="fill") # Stacked bar chart
p1 <- p1 + ggtitle("Gender by Next Month Default") + xlab("") + ylab("NEXT_MONTH_DEFAULT") # Adds title
p1 <- p1 + facet_grid(facets=. ~ NEXT_MONTH_DEFAULT) # Side by side bar chart
p1 <- p1 + coord_polar(theta="y") # side by side pie chart
p1
```



As it can be clearly seen here, gender does not have a significant difference when compared with defaulting credit cards and not defaulting.

Marital status

```
train_df$Gender <- factor(train_df$MARITAL_STATUS) # converts to a categorical variable
train_df$NEXT_MONTH_DEFAULT <- factor(train_df$NEXT_MONTH_DEFAULT) # converts to a categorical variable
p2 <- ggplot(data=train_df, aes(x=factor(1), stat="bin", fill=MARITAL_STATUS)) +
  geom_bar(position="fill") # Stacked bar chart
p2 <- p2 + ggtitle("Marital Status by Next Month Default") + xlab("") + ylab("NEXT_MONTH_DEFAULT") # Ad
p2 <- p2 + facet_grid(facets=. ~ NEXT_MONTH_DEFAULT) # Side by side bar chart
p2 <- p2 + coord_polar(theta="y") # side by side pie chart
p2
```



As it can be clearly seen here, marital status does not have a significant difference when compared with defaulting credit cards and not defaulting. -*Hence we have clear reasons for removing the variables gender and marital status_*

Education Status

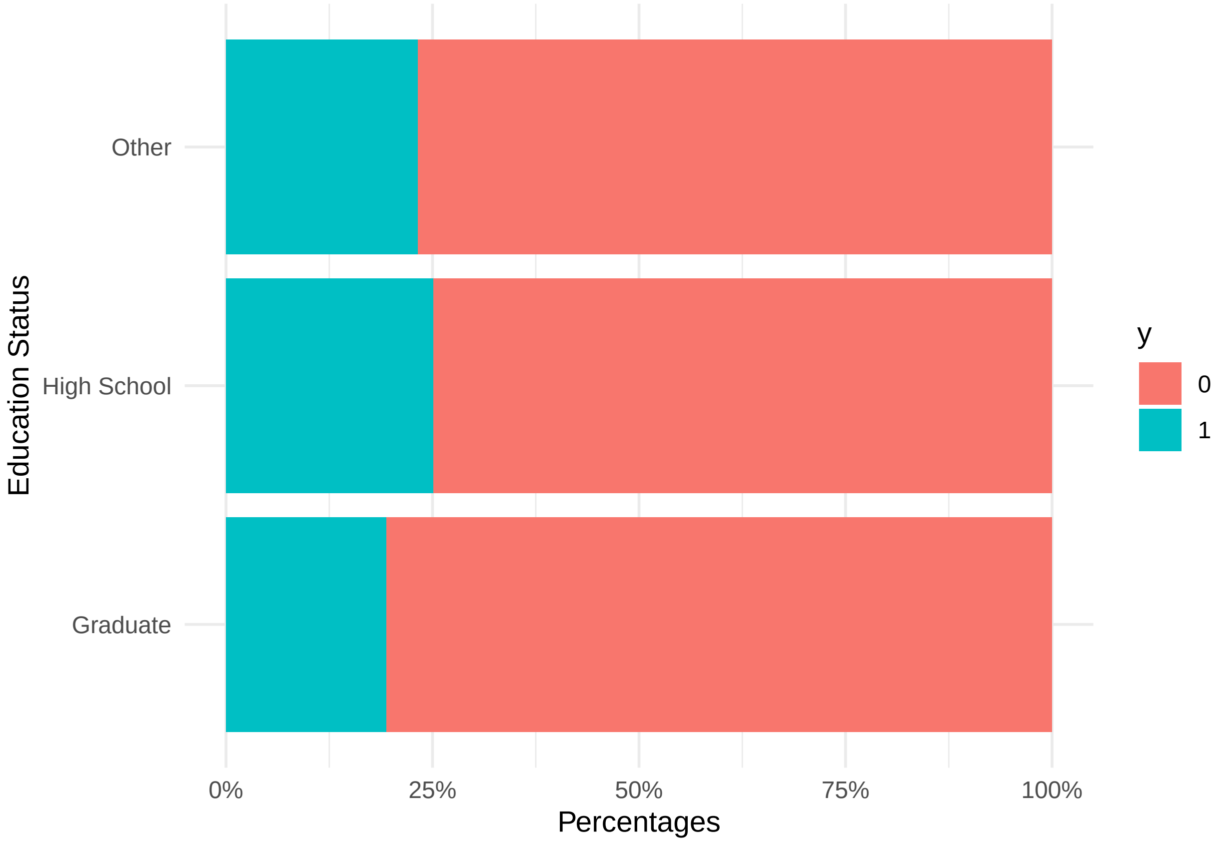
```
education_response_ct <- CrossTable(train_df$EDUCATION_STATUS,
                               train_df$NEXT_MONTH_DEFAULT ,
                               prop.r= FALSE,
                               prop.c=TRUE,
                               prop.chisq=FALSE,
                               chisq = TRUE,
                               prop.t=TRUE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  24000
##
##
##                           | train_df$NEXT_MONTH_DEFAULT
## train_df$EDUCATION_STATUS |         0 |         1 | Row Total |
## --------------------------|-----------|-----------|-----------|
##                  Graduate |      6828 |      1650 |      8478 |
##                           |     0.366 |     0.310 |           |
##                           |     0.284 |     0.069 |           |
## --------------------------|-----------|-----------|-----------|
##               High School |      2939 |       986 |      3925 |
##                           |     0.157 |     0.185 |           |
```

```
##                           |      0.122 |      0.041 |            |
## --------------------------|------------|------------|------------|
##                     Other |       8903 |       2694 |      11597 |
##                           |      0.477 |      0.505 |            |
##                           |      0.371 |      0.112 |            |
## --------------------------|------------|------------|------------|
##              Column Total |      18670 |       5330 |      24000 |
##                           |      0.778 |      0.222 |            |
## --------------------------|------------|------------|------------|
##
##
## Statistics for All Table Factors
##
##
## Pearson's Chi-squared test
## ------------------------------------------------------------
## Chi^2 =  63.292     d.f. =  2     p =  1.804334e-14
##
##
##
```

Plotting

```
ggplot(data=as.data.frame(education_response_ct$prop.tbl),
       aes(x=x,y=Freq,fill=y))+
  geom_bar(stat = "identity",position = "fill")+
  scale_y_continuous(labels = scales::percent_format())+
  coord_flip()+theme_minimal()+
  xlab("Education Status")+ylab("Percentages")
```



Even though it is not evident that there is a relationship between education status and the response variable, the chi square test confirms that there is an association between the two variables ### Balance Limit

```
blv <- train_df%>%
  mutate(Balance_Limit_V1=trimws(Balance_Limit_V1)) %>%
  subset(select=Balance_Limit_V1)
train_df$Balance_Limit_V1 <- factor(as.factor(blv$Balance_Limit_V1),
                          levels=c("100K","200K","300K","400K","500K","1M","1.5M","2.5M"))
balance_response_ct <- CrossTable(train_df$Balance_Limit_V1 ,
                               train_df$NEXT_MONTH_DEFAULT ,
                               prop.r= FALSE,
                               prop.c=TRUE,
                               prop.chisq=FALSE,
                               chisq = TRUE,
                               prop.t=TRUE)
```

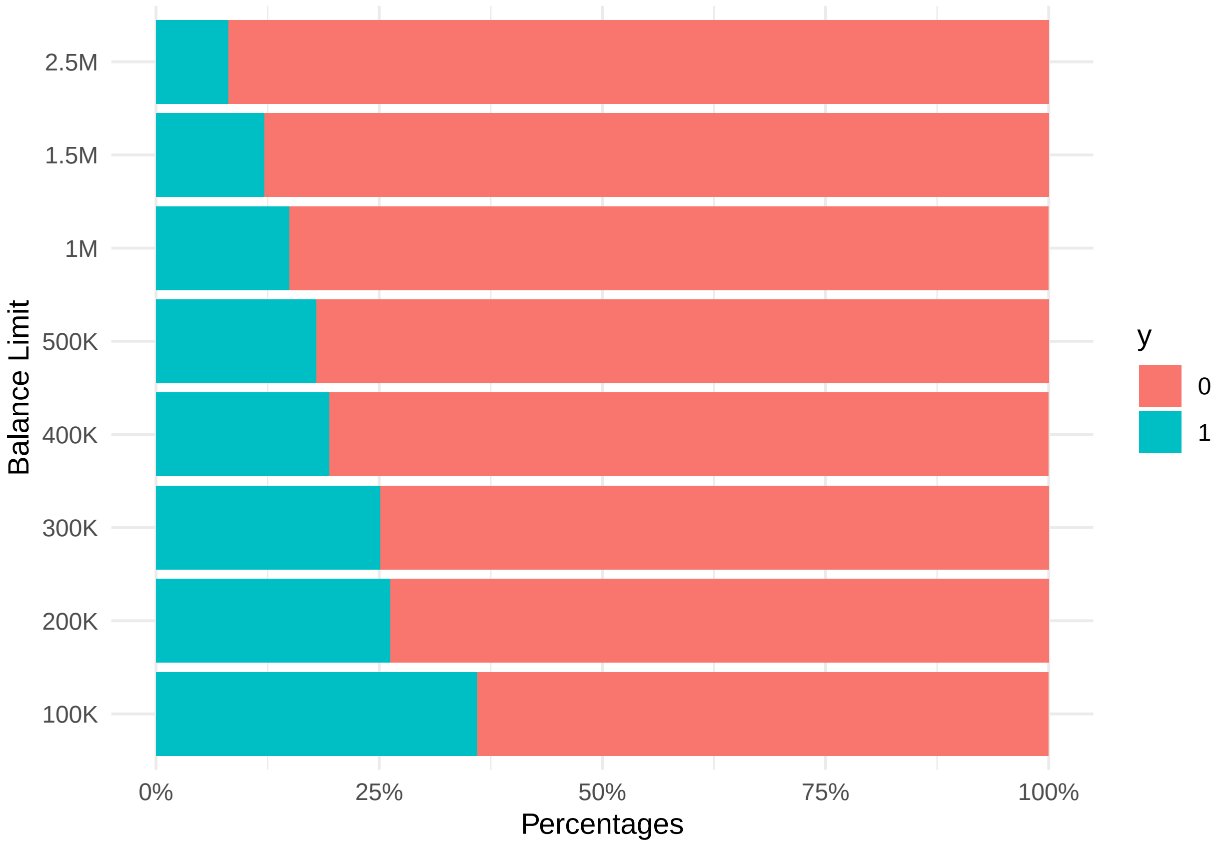
```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  24000
##
```

```
##
##                             | train_df$NEXT_MONTH_DEFAULT
## train_df$Balance_Limit_V1 |         0 |         1 | Row Total |
## --------------------------|-----------|-----------|-----------|
##                      100K |      2207 |      1242 |      3449 |
##                           |     0.118 |     0.233 |           |
##                           |     0.092 |     0.052 |           |
## --------------------------|-----------|-----------|-----------|
##                      200K |      3805 |      1354 |      5159 |
##                           |     0.204 |     0.254 |           |
##                           |     0.159 |     0.056 |           |
## --------------------------|-----------|-----------|-----------|
##                      300K |      1805 |       606 |      2411 |
##                           |     0.097 |     0.114 |           |
##                           |     0.075 |     0.025 |           |
## --------------------------|-----------|-----------|-----------|
##                      400K |      2469 |       596 |      3065 |
##                           |     0.132 |     0.112 |           |
##                           |     0.103 |     0.025 |           |
## --------------------------|-----------|-----------|-----------|
##                      500K |      2289 |       501 |      2790 |
##                           |     0.123 |     0.094 |           |
##                           |     0.095 |     0.021 |           |
## --------------------------|-----------|-----------|-----------|
##                        1M |      5061 |       890 |      5951 |
##                           |     0.271 |     0.167 |           |
##                           |     0.211 |     0.037 |           |
## --------------------------|-----------|-----------|-----------|
##                      1.5M |      1000 |       138 |      1138 |
##                           |     0.054 |     0.026 |           |
##                           |     0.042 |     0.006 |           |
## --------------------------|-----------|-----------|-----------|
##                      2.5M |        34 |         3 |        37 |
##                           |     0.002 |     0.001 |           |
##                           |     0.001 |     0.000 |           |
## --------------------------|-----------|-----------|-----------|
##              Column Total |     18670 |      5330 |     24000 |
##                           |     0.778 |     0.222 |           |
## --------------------------|-----------|-----------|-----------|
##
##
## Statistics for All Table Factors
##
##
## Pearson's Chi-squared test
## ------------------------------------------------------------
## Chi^2 =  736.0712     d.f. =  7     p =  1.148874e-154
##
##
##
```

Plotting

```
ggplot(data=as.data.frame(balance_response_ct$prop.tbl),
       aes(fill=y,y=Freq,x=x))+
```

```r
  geom_bar(stat = "identity",position = "fill")+
  scale_y_continuous(labels = scales::percent_format())+
  coord_flip()+theme_minimal()+
  xlab("Balance Limit")+ylab("Percentages")
```

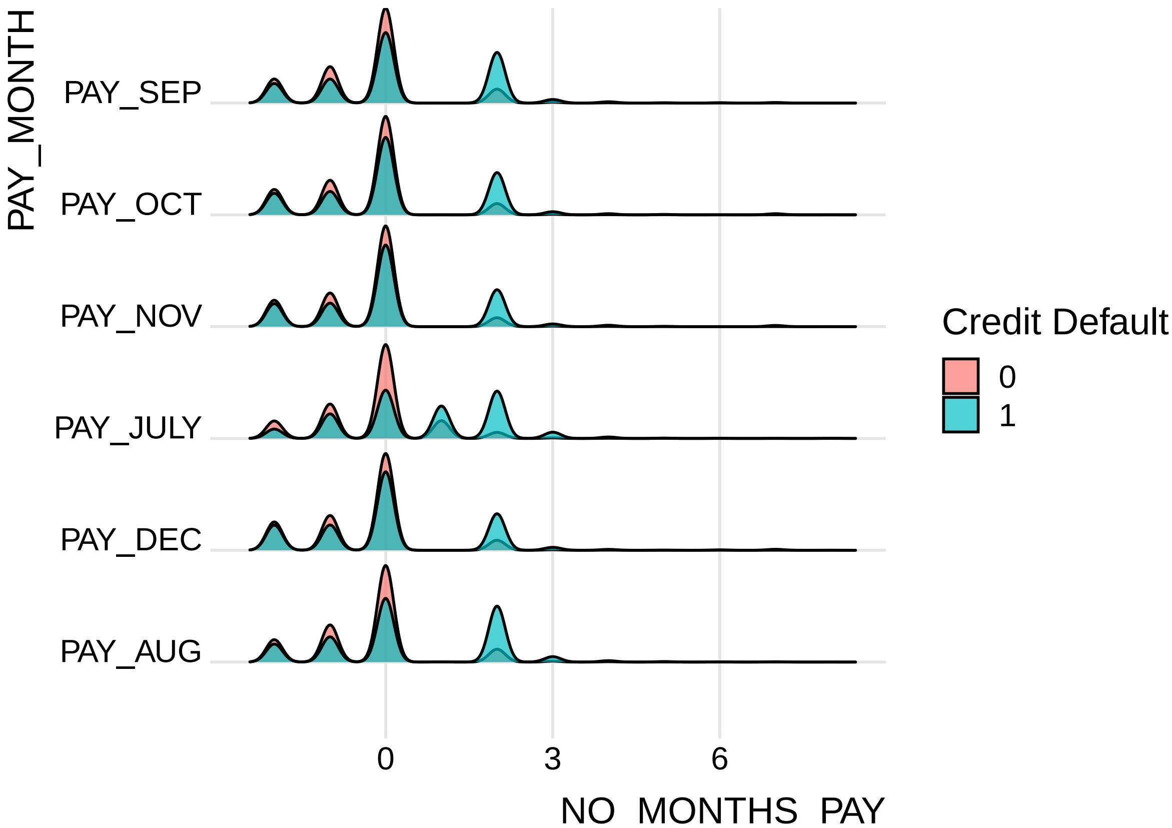


Here it is clear that a client with a lower balance limit has a higher chance of getting a credit card default. Also from the chi square test it is clear that with a p-value less than alpha=0.05 we can reject H0 and come to the conclusion that there is balance limit and the response variable are not independent

Payment due variable

```r
ggplot(train_df %>%
  gather(PAY_JULY , PAY_AUG , PAY_SEP , PAY_OCT , PAY_NOV , PAY_DEC , key = "PAY_MONTH",value = "NO_MON
  geom_density_ridges(scale=0.9,alpha=0.7) +
  theme_ridges()+
  labs(fill='Credit Default')
```

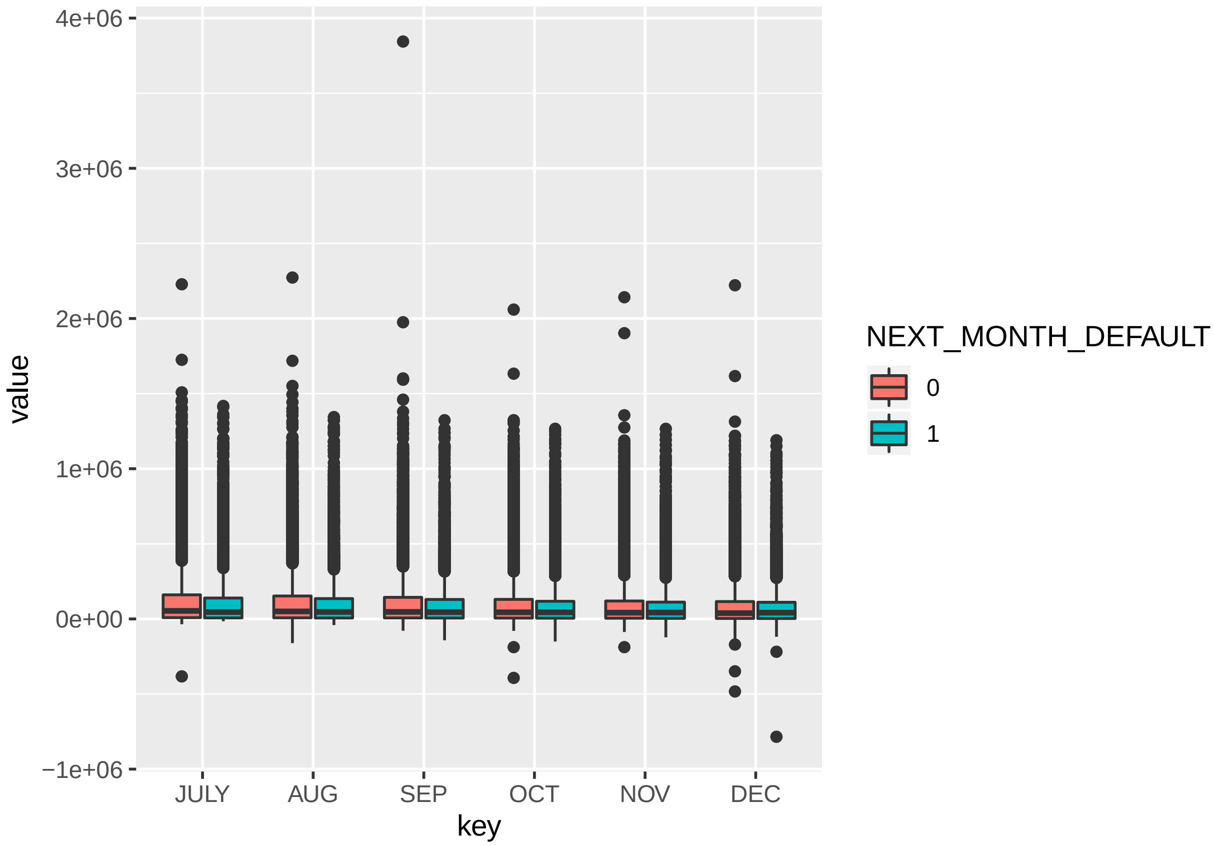
```
## Picking joint bandwidth of 0.146
```



Quantitative Variables

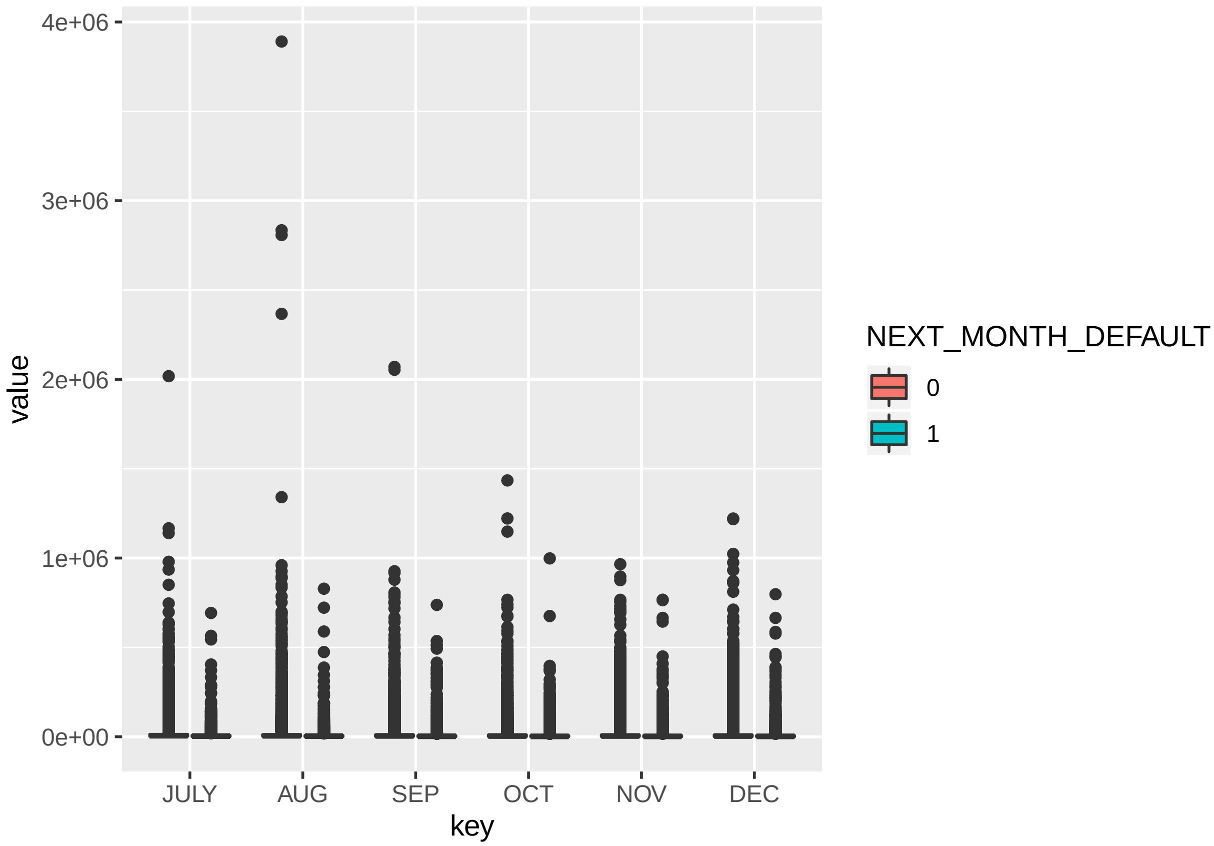
Due amounts

```
ggplot(data=gather(train_df,key,value,c("DUE_AMT_JULY","DUE_AMT_AUG",
                                        "DUE_AMT_SEP","DUE_AMT_OCT",
                                        "DUE_AMT_NOV","DUE_AMT_DEC")) %>%
         mutate(key = factor(key,levels=c("DUE_AMT_JULY","DUE_AMT_AUG",
                                        "DUE_AMT_SEP","DUE_AMT_OCT",
                                        "DUE_AMT_NOV","DUE_AMT_DEC"))),
       aes(x = key,y = value,fill=NEXT_MONTH_DEFAULT))+
  geom_boxplot()+
  scale_x_discrete(labels=c("JULY","AUG","SEP","OCT","NOV","DEC"))
```



Paid amounts

```
ggplot(data=gather(train_df,key,value,c("PAID_AMT_JULY","PAID_AMT_AUG",
                                       "PAID_AMT_SEP","PAID_AMT_OCT",
                                       "PAID_AMT_NOV","PAID_AMT_DEC")) %>%
         mutate(key = factor(key,levels=c("PAID_AMT_JULY","PAID_AMT_AUG",
                                       "PAID_AMT_SEP","PAID_AMT_OCT",
                                       "PAID_AMT_NOV","PAID_AMT_DEC"))),
       aes(x = key,y = value,fill=NEXT_MONTH_DEFAULT))+
  geom_boxplot()+
  scale_x_discrete(labels=c("JULY","AUG","SEP","OCT","NOV","DEC"))
```



Further visualizations

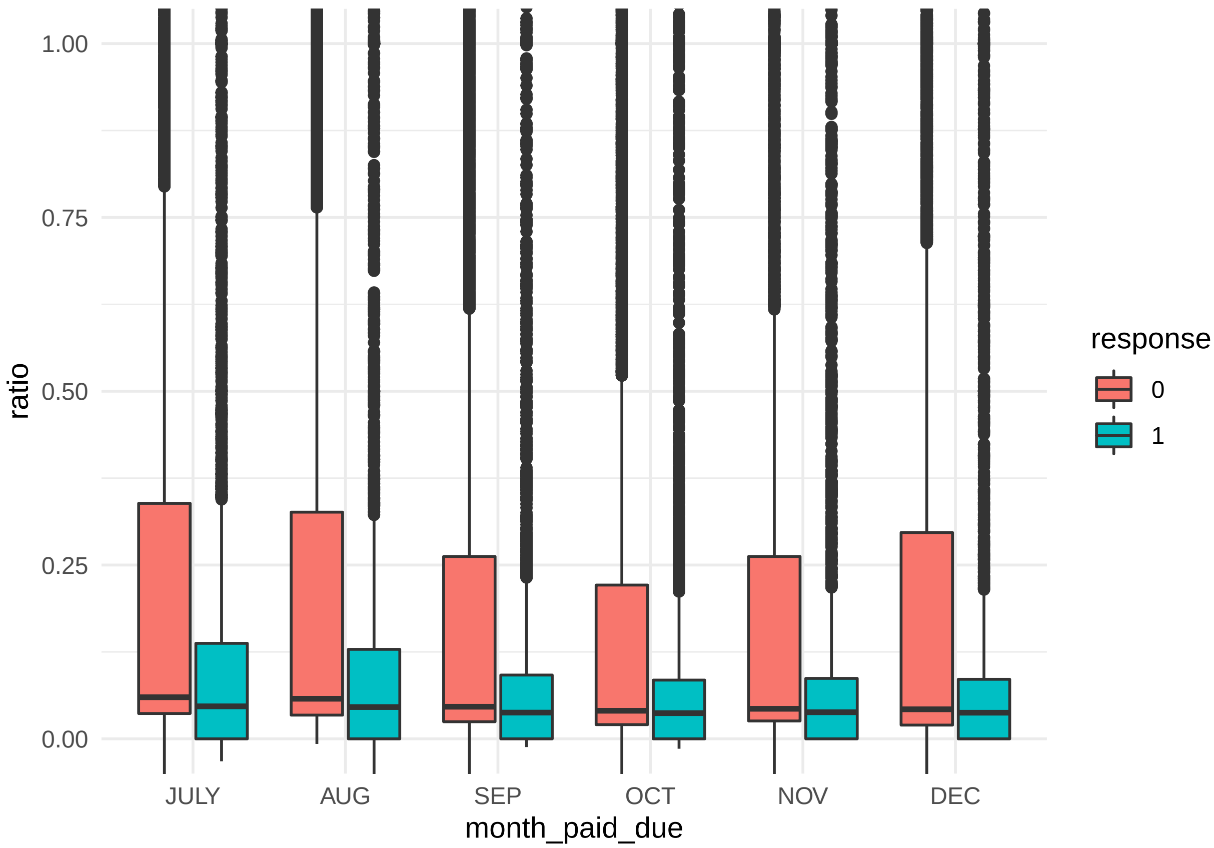
Paid amount as a ratio of due amount

```
eps<-0.1
train_df$duepaid_JULY <- train_df$PAID_AMT_JULY/(train_df$DUE_AMT_JULY+eps)
train_df$duepaid_AUG <-  train_df$PAID_AMT_AUG/(train_df$DUE_AMT_AUG+eps)
train_df$duepaid_SEP <-  train_df$PAID_AMT_SEP/(train_df$DUE_AMT_SEP+eps)
train_df$duepaid_OCT <-  train_df$PAID_AMT_OCT/(train_df$DUE_AMT_OCT+eps)
train_df$duepaid_NOV <-  train_df$PAID_AMT_NOV/(train_df$DUE_AMT_NOV+eps)
train_df$duepaid_DEC <-  train_df$PAID_AMT_DEC/(train_df$DUE_AMT_DEC+eps)

train_df$duepaid_JULY[is.nan(train_df$duepaid_JULY)] <- 0
train_df$duepaid_AUG[is.nan(train_df$duepaid_AUG)] <- 0
train_df$duepaid_SEP[is.nan(train_df$duepaid_SEP)] <- 0
train_df$duepaid_OCT[is.nan(train_df$duepaid_OCT)] <- 0
train_df$duepaid_NOV[is.nan(train_df$duepaid_NOV)] <- 0
train_df$duepaid_DEC[is.nan(train_df$duepaid_DEC)] <- 0

new04 <- data.frame(train_df$duepaid_NOV,train_df$duepaid_OCT,
                    train_df$duepaid_SEP,train_df$duepaid_AUG,
                    train_df$duepaid_JULY,train_df$duepaid_DEC,
                    as.factor(train_df$NEXT_MONTH_DEFAULT))
new05 <- new04 %>%
  gather(train_df.duepaid_DEC,train_df.duepaid_NOV,
         train_df.duepaid_OCT,train_df.duepaid_SEP,
         train_df.duepaid_AUG,train_df.duepaid_JULY, key = "month_paid_due",value = "ratio")
```

```
new06 <- na.omit(new05)
new06$month_paid_due <- factor(new06$month_paid_due,
                               levels=c("train_df.duepaid_JULY","train_df.duepaid_AUG",
                                        "train_df.duepaid_SEP","train_df.duepaid_OCT",
                                        "train_df.duepaid_NOV","train_df.duepaid_DEC"))
colnames(new06)[1] <- "response"
ggplot(new06, aes(x = month_paid_due , y = ratio)) +
  geom_boxplot(aes(fill=response))+
  coord_cartesian(ylim=c(0,1))+
  theme_minimal()+
  scale_x_discrete(labels=c("JULY","AUG","SEP","OCT","NOV","DEC"))
```

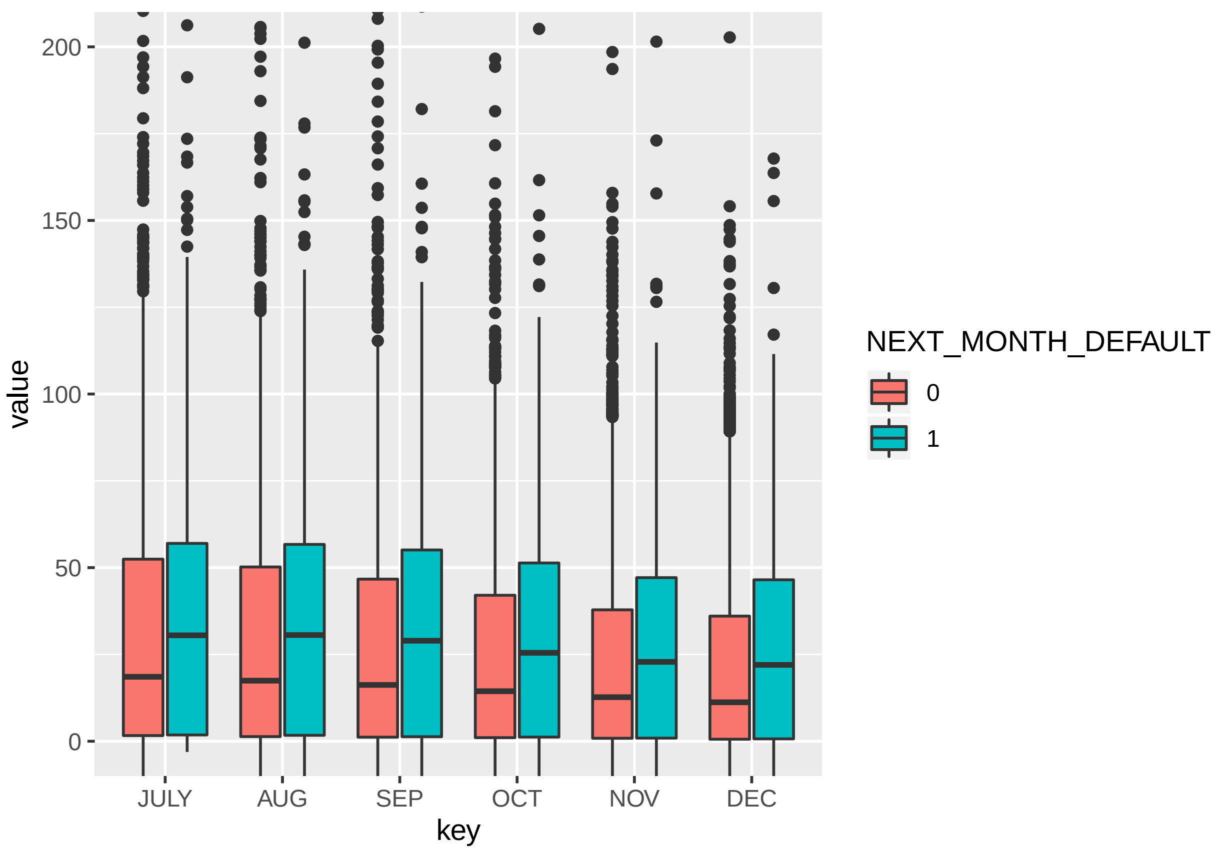


Due amount as a ratio of credit limit

```
train_df <- train_df %>% mutate(
  Balance_Credit_Limit_Numeric = case_when(
    Balance_Limit_V1 == "100K" ~ 100000,
    Balance_Limit_V1 == "200K" ~ 200000,
    Balance_Limit_V1 == "300K" ~ 300000,
    Balance_Limit_V1 == "400K" ~ 400000,
    Balance_Limit_V1 == "500K" ~ 500000,
    Balance_Limit_V1 == "1M" ~ 1000000,
    Balance_Limit_V1 == "1.5M" ~ 1500000,
    Balance_Limit_V1 == "2M" ~ 2000000,
    Balance_Limit_V1 == "2.5M" ~ 2500000
  )
) %>%
```

```r
  mutate(
    Due_Credit_Lim_JULY=(DUE_AMT_JULY/Balance_Credit_Limit_Numeric) * 100,
    Due_Credit_Lim_AUG=(DUE_AMT_AUG/Balance_Credit_Limit_Numeric) * 100,
    Due_Credit_Lim_SEP=(DUE_AMT_SEP/Balance_Credit_Limit_Numeric) * 100,
    Due_Credit_Lim_OCT=(DUE_AMT_OCT/Balance_Credit_Limit_Numeric) * 100,
    Due_Credit_Lim_NOV=(DUE_AMT_NOV/Balance_Credit_Limit_Numeric) * 100,
    Due_Credit_Lim_DEC=(DUE_AMT_DEC/Balance_Credit_Limit_Numeric) * 100
  ) %>%
  na.omit()
```

```r
ggplot(data=gather(train_df,key,value,c("Due_Credit_Lim_JULY","Due_Credit_Lim_AUG",
                                        "Due_Credit_Lim_SEP","Due_Credit_Lim_OCT",
                                        "Due_Credit_Lim_NOV","Due_Credit_Lim_DEC")) %>%
         mutate(key = factor(key,levels=c("Due_Credit_Lim_JULY","Due_Credit_Lim_AUG",
                                        "Due_Credit_Lim_SEP","Due_Credit_Lim_OCT",
                                        "Due_Credit_Lim_NOV","Due_Credit_Lim_DEC"))),
       aes(x=key,y=value))+
  geom_boxplot(aes(fill=NEXT_MONTH_DEFAULT))+
  coord_cartesian(ylim=c(0,200))+
  scale_x_discrete(labels=c("JULY","AUG","SEP","OCT","NOV","DEC"))
```



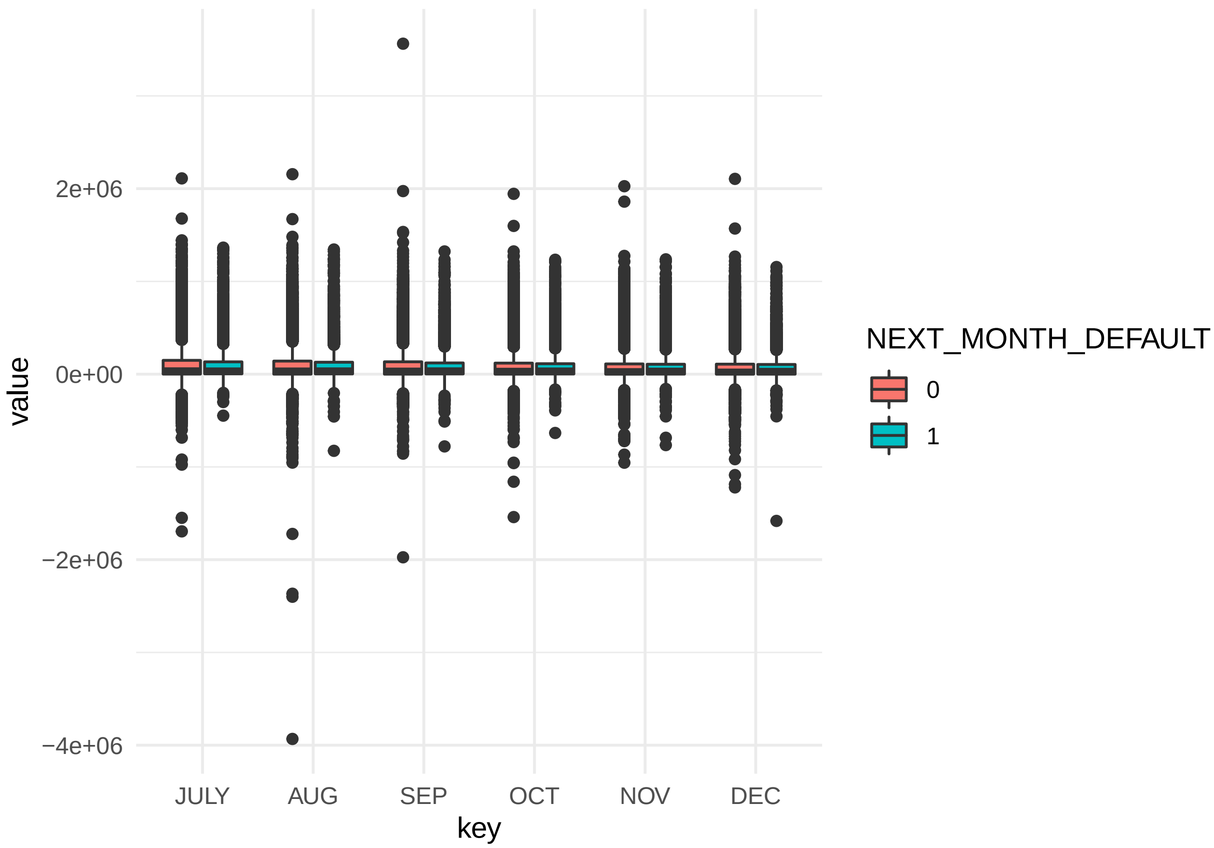
Paid amount as a ratio of credit limit

Balance feature

Here we consider the balance as the `Due amount - Paid amount`

```
train_df$balance_july <- train_df$DUE_AMT_JULY - train_df$PAID_AMT_JULY
train_df$balance_aug <- train_df$DUE_AMT_AUG - train_df$PAID_AMT_AUG
train_df$balance_sep <- train_df$DUE_AMT_SEP - train_df$PAID_AMT_SEP
train_df$balance_oct <- train_df$DUE_AMT_OCT - train_df$PAID_AMT_OCT
train_df$balance_nov <- train_df$DUE_AMT_NOV - train_df$PAID_AMT_NOV
train_df$balance_dec <- train_df$DUE_AMT_DEC - train_df$PAID_AMT_DEC

ggplot(data=gather(train_df,key,value,c("balance_july","balance_aug",
                                        "balance_sep","balance_oct",
                                        "balance_nov","balance_dec")) %>%
         mutate(key = factor(key,levels =c("balance_july","balance_aug",
                                        "balance_sep","balance_oct",
                                        "balance_nov","balance_dec"))),
       aes(x = key,y = value))+
  geom_boxplot(aes(fill=NEXT_MONTH_DEFAULT))+
  theme_minimal()+
  scale_x_discrete(labels=c("JULY","AUG","SEP","OCT","NOV","DEC"))
```

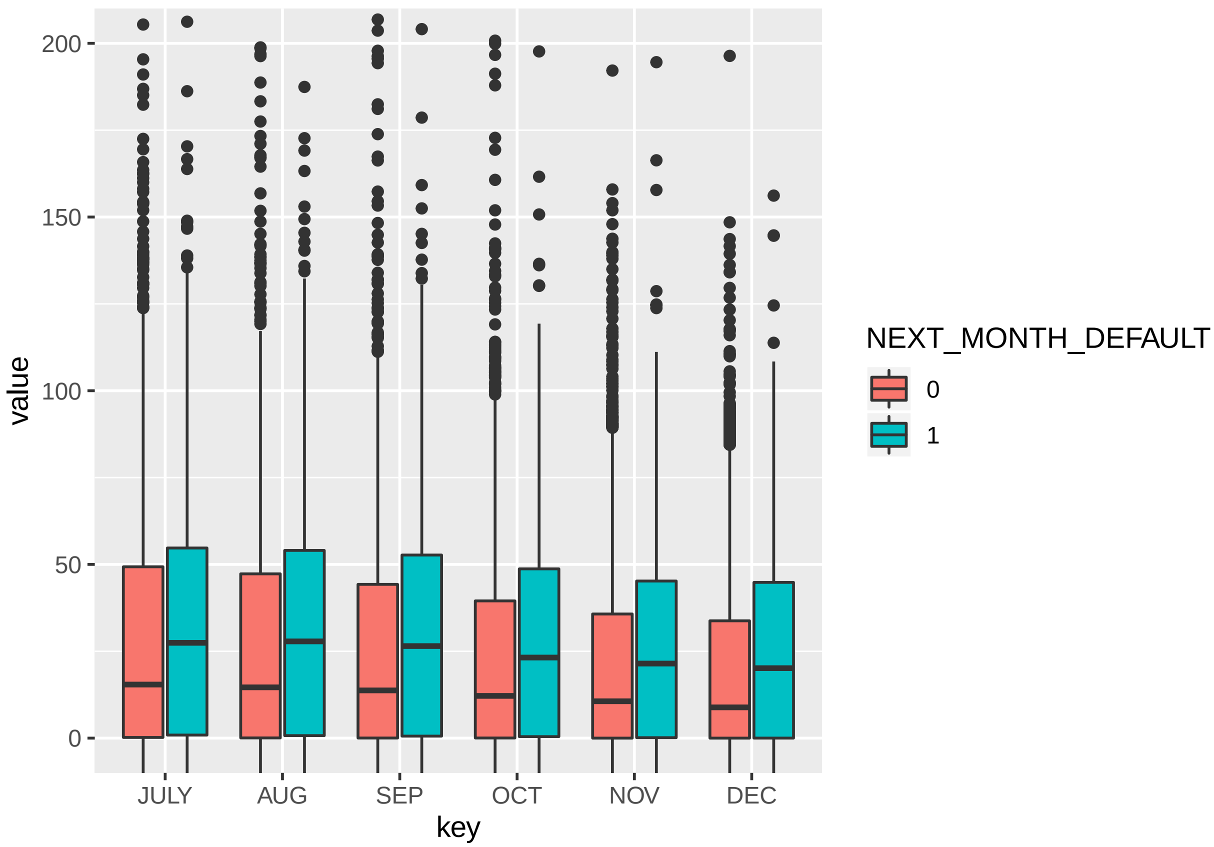


Balance as a ratio of of credit limit

```
train_df <- train_df %>% mutate(
    balance_Credit_Lim_JULY=(balance_july/Balance_Credit_Limit_Numeric) * 100,
    balance_Credit_Lim_AUG=(balance_aug/Balance_Credit_Limit_Numeric) * 100,
    balance_Credit_Lim_SEP=(balance_sep/Balance_Credit_Limit_Numeric) * 100,
    balance_Credit_Lim_OCT=(balance_oct/Balance_Credit_Limit_Numeric) * 100,
    balance_Credit_Lim_NOV=(balance_nov/Balance_Credit_Limit_Numeric) * 100,
    balance_Credit_Lim_DEC=(balance_dec/Balance_Credit_Limit_Numeric) * 100
  ) %>%
```

```
  na.omit()
```

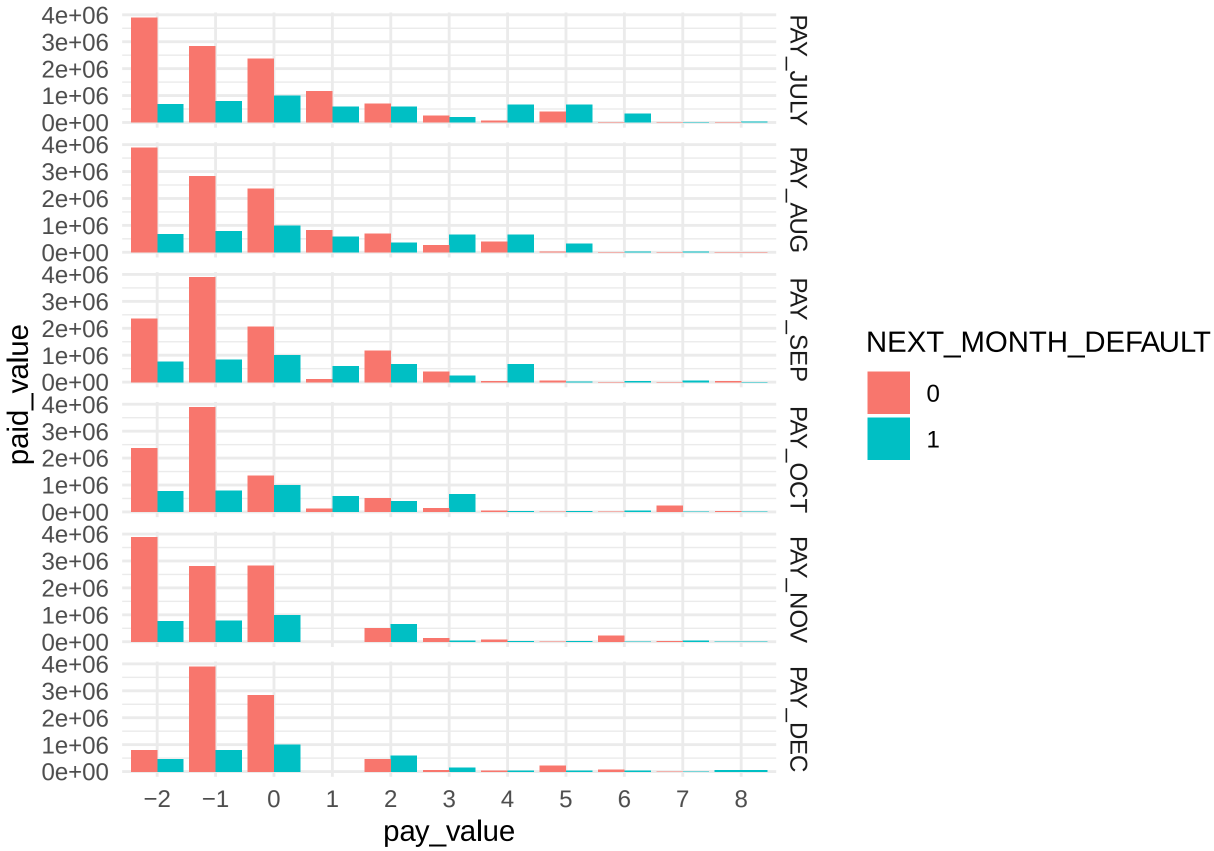
```
ggplot(data=gather(train_df,key,value,c("balance_Credit_Lim_JULY","balance_Credit_Lim_AUG",
                                        "balance_Credit_Lim_SEP","balance_Credit_Lim_OCT",
                                        "balance_Credit_Lim_NOV","balance_Credit_Lim_DEC")) %>%
         mutate(key = factor(key,levels=c("balance_Credit_Lim_JULY","balance_Credit_Lim_AUG",
                                        "balance_Credit_Lim_SEP","balance_Credit_Lim_OCT",
                                        "balance_Credit_Lim_NOV","balance_Credit_Lim_DEC"))),
       aes(x=key,y=value))+
  geom_boxplot(aes(fill=NEXT_MONTH_DEFAULT))+
  coord_cartesian(ylim=c(0,200))+
  scale_x_discrete(labels=c("JULY","AUG","SEP","OCT","NOV","DEC"))
```



Pay delay with the Paid value

```
ggplot(data=train_df%>%
         gather(pay_key,pay_value,c("PAY_JULY","PAY_AUG",
                                        "PAY_SEP","PAY_OCT",
                                        "PAY_NOV","PAY_DEC")) %>%
         gather(paid_key,paid_value,c("PAID_AMT_JULY","PAID_AMT_AUG",
                                      "PAID_AMT_SEP","PAID_AMT_OCT",
                                      "PAID_AMT_NOV","PAID_AMT_DEC")) %>%
         mutate(pay_key = factor(pay_key,levels=c("PAY_JULY","PAY_AUG",
                                        "PAY_SEP","PAY_OCT",
                                        "PAY_NOV","PAY_DEC")),
                pay_value=factor(pay_value,levels=seq(-2,9))),
       aes(y=paid_value,x=pay_value,fill=NEXT_MONTH_DEFAULT))+
  # geom_line(aes(color=NEXT_MONTH_DEFAULT))+
```

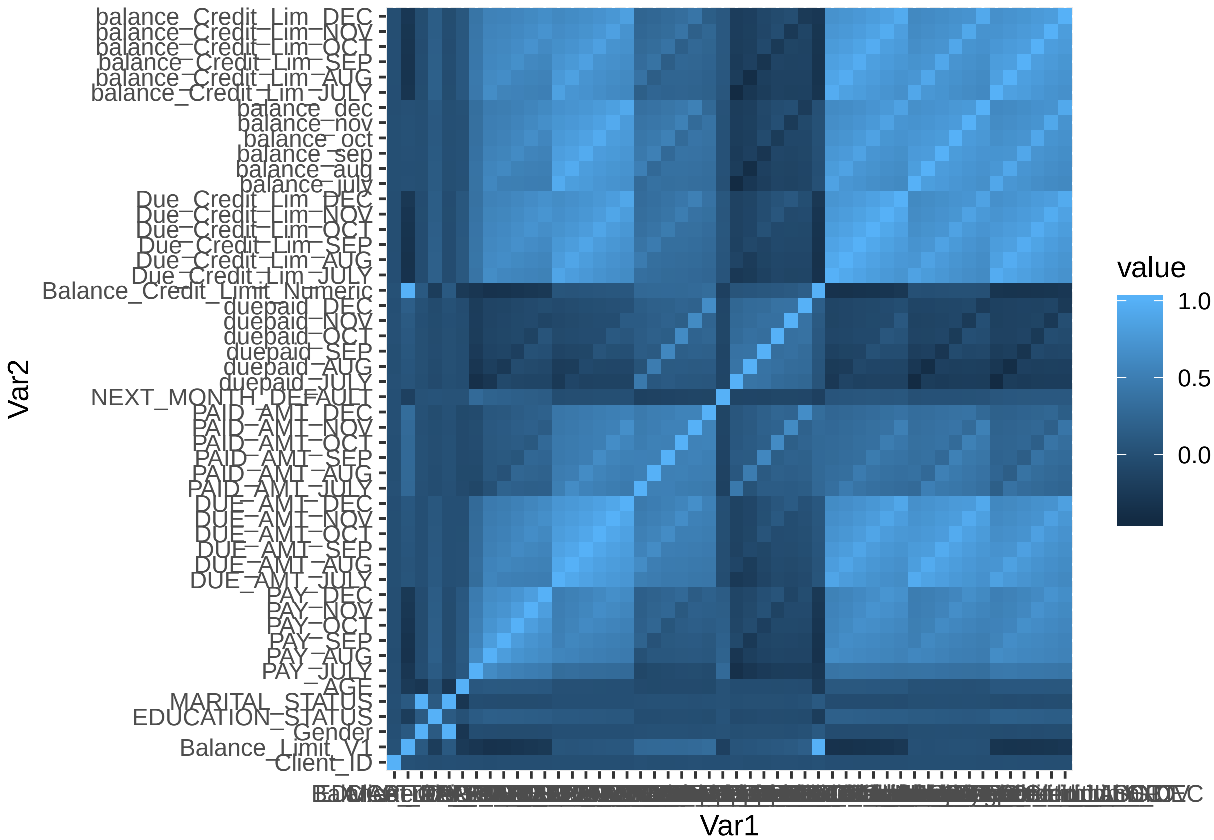
```
  geom_bar(position="dodge", stat="identity")+
  facet_grid(pay_key ~ .) + theme_minimal()
```



Correlation matrix plot

Disregard this as it is too cluttered

```
train_df_to_cor <- train_df %>% mutate_if(is.factor,as.numeric)
corrmat <- cor(train_df_to_cor,method="spearman")
melted <- melt(corrmat) %>%
  mutate(text = paste0("x: ", Var1, "\n", "y: ", Var2, "\n", "Value: ",round(value,2), "\n"))
p <- ggplot(melted, aes(Var1, Var2, fill= value, text=text)) +
  geom_tile()
p
```




```
# ggplotly(p, tooltip="text")
```