# Report_bank_english

Felipe Carvalhal Moitinho

05/12/2020

# Introduction

The database used is of cases of default payments in Taiwan. The database has 23 variables ranging from personal data such as gender, age, education level and marital status to financial data related to debt such as the amount of credit assigned, payments paid and the value of the invoice.
The objective of the work is to test the ability of the algorithm to predict whether or not the customer will make the payment in the month. For this, the model will be estimated by two different methodologies. First, it will be manipulated to factor out qualitative variables. Subsequently, some variables from the database will be analyzed to identify the variables that will be used in the algorithms. After identifying the variables that will compose the equation, the calculated results will be exposed and analyzed. Finally, the concusions and limitations of the work will be exposed.
The work consists of 3 parts in addition to this introduction:
* MEthods / Analysys: will contain the steps for cleaning the data standardization as well as the analysis and methodologies;
* Results: will contain the results and the discussion of the results; and
* Conclusion and limitation.

# Methods/Analysis

In this chapter, some variables of the database will be analyzed and their behavior in relation to the default or not defaut situation. First the database will be worked and later the variables will be analyzed.
To adjust the database the variables SEX, EDUCATION, MARRIAGE AND default payment next month were factored using the `factor` function. The variables `default payment next month` and `LIMIT_BAL` have been renamed to facilitate interpretation.
Subsequently, the database was filtered to eliminate all observations that had NA. After performing these operations, the database was analyzed. ## Data analysis and visualization

```
bank$SEX <- factor(bank$SEX, levels = c(1,2), labels = c("Male", "Female"))
bank$EDUCATION <- factor(bank$EDUCATION, levels = c(1,2,3,4), labels = c("graduat
e","university","high_school", "others"))
bank$MARRIAGE <- factor(bank$MARRIAGE, levels = c(1,2,3), labels = c("Married", "S
ingle", "Others"))
bank$`default payment next month` <- factor(bank$`default payment next month`, lev
els = c(0,1), labels = c("Not default","default"))
bank <- rename(bank, default_payment_next_month = `default payment next month`, cr
edit_amount = LIMIT_BAL)
bank <- select(bank, c(-(PAY_4:PAY_6), -(BILL_AMT4:BILL_AMT6), -(PAY_AMT4:PAY_AMT
6)))
bank <- filter(bank, !is.na(credit_amount) & !is.na(SEX) & !is.na(EDUCATION) & !i
s.na(MARRIAGE) & !is.na(AGE) & !is.na(PAY_0) & !is.na(PAY_2) & !is.na(PAY_3) & !i
s.na(BILL_AMT1) & !is.na(BILL_AMT2) & !is.na(BILL_AMT3) & !is.na(default_payment_n
ext_month))
```

Observing the first table, it is noted that the average age of the observations is 35 years. The youngest is

21 and the oldest 79 years old.

```
as.table(summary(bank$AGE))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   21.00   28.00   34.00   35.46   41.00   79.00
```

In the amounts of credit assigned, they range from $ 10,000 to $ 1,000,000. The average is $ 140000.

```
summary(bank$credit_amount)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   10000   50000  140000  167551  240000 1000000
```

Tables 1 shows that the vast majority of observations are divided between married or single. Ta bullet 2 shows that most of the observations are female. In Graph 1.1, it is noted that the male gender has a slightly higher proportion of default individuals than the female. In Graph 1.2, cases of default are divided according to marital status. For the case of marital status, singles are the ones with the lowest proportion of defaulters. followed by married and other situations. Graph 1.3 shows that the higher the level of education, the greater the proportion of people in default. Finally, in the case of age, the following movement can be seen in the protection of defaulters: when young, the proportion of defaulters is relatively high, but this proportion is reduced up to approximately 40 years. After 40 years of age, the index increases satisfactorily with peaks such as 60 and after 70.

```
kable(table(bank$MARRIAGE), col.names = c("Marriage", "Freq"), caption = "Table 1:
Marriage")
```

Table 1: Marriage

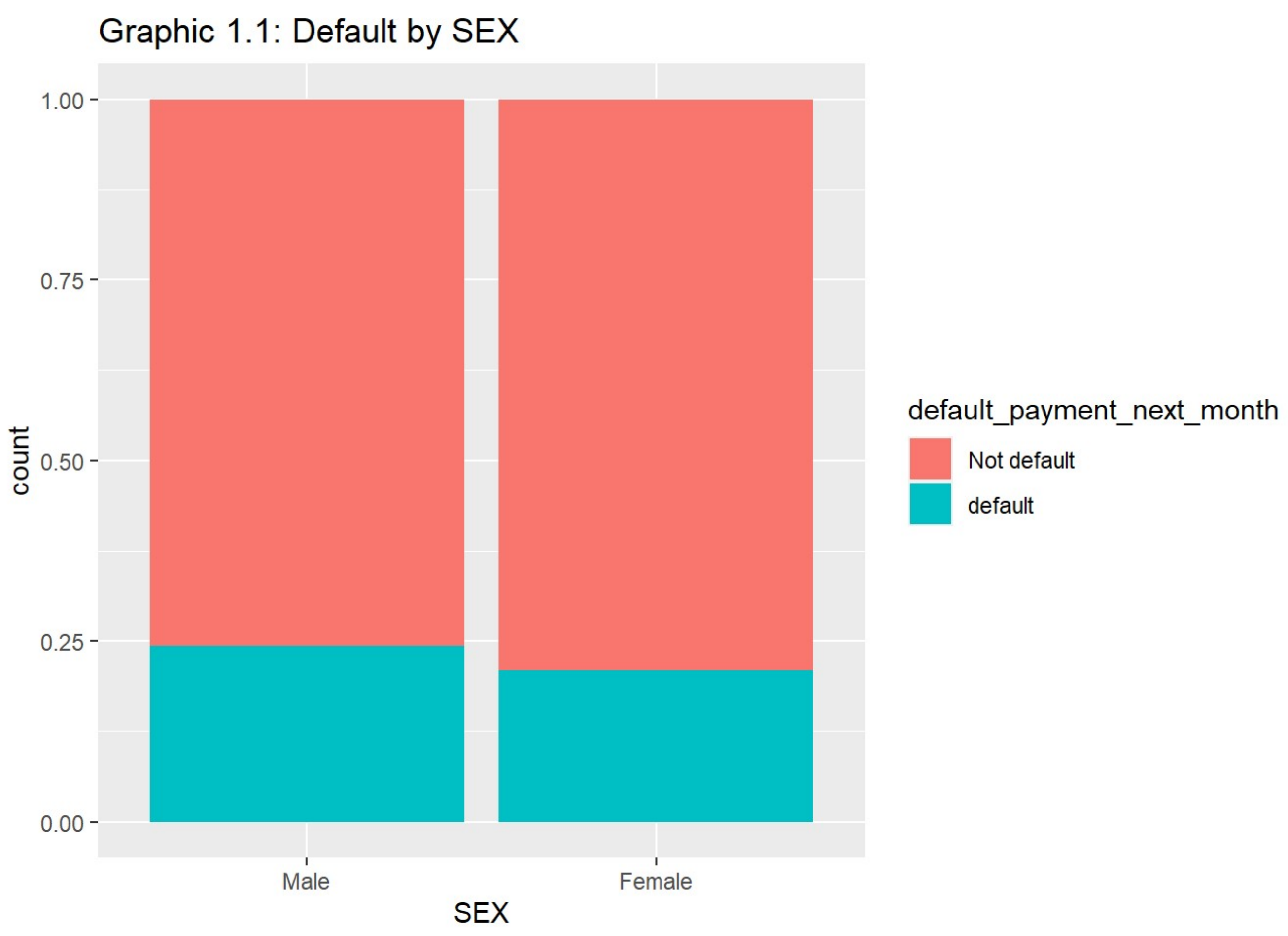
| Marriage | Freq |
|---|---:|
| Married | 13477 |
| Single | 15806 |
| Others | 318 |

```
kable(table(bank$SEX), col.names = c("Sex", "Freq"), caption = "Table 2: Sex")
```

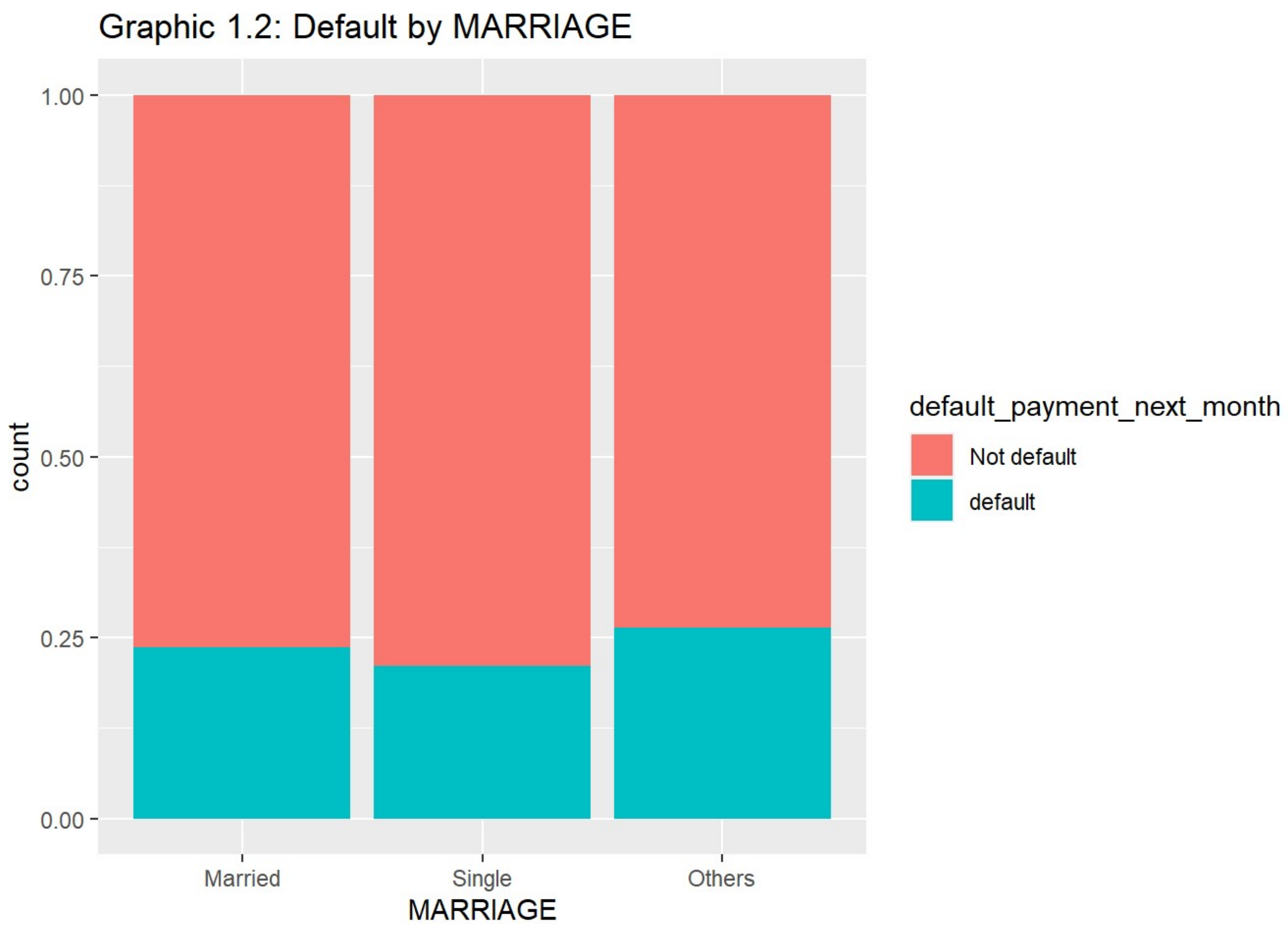
Table 2: Sex

| Sex | Freq |
|---|---:|
| Male | 11746 |
| Female | 17855 |

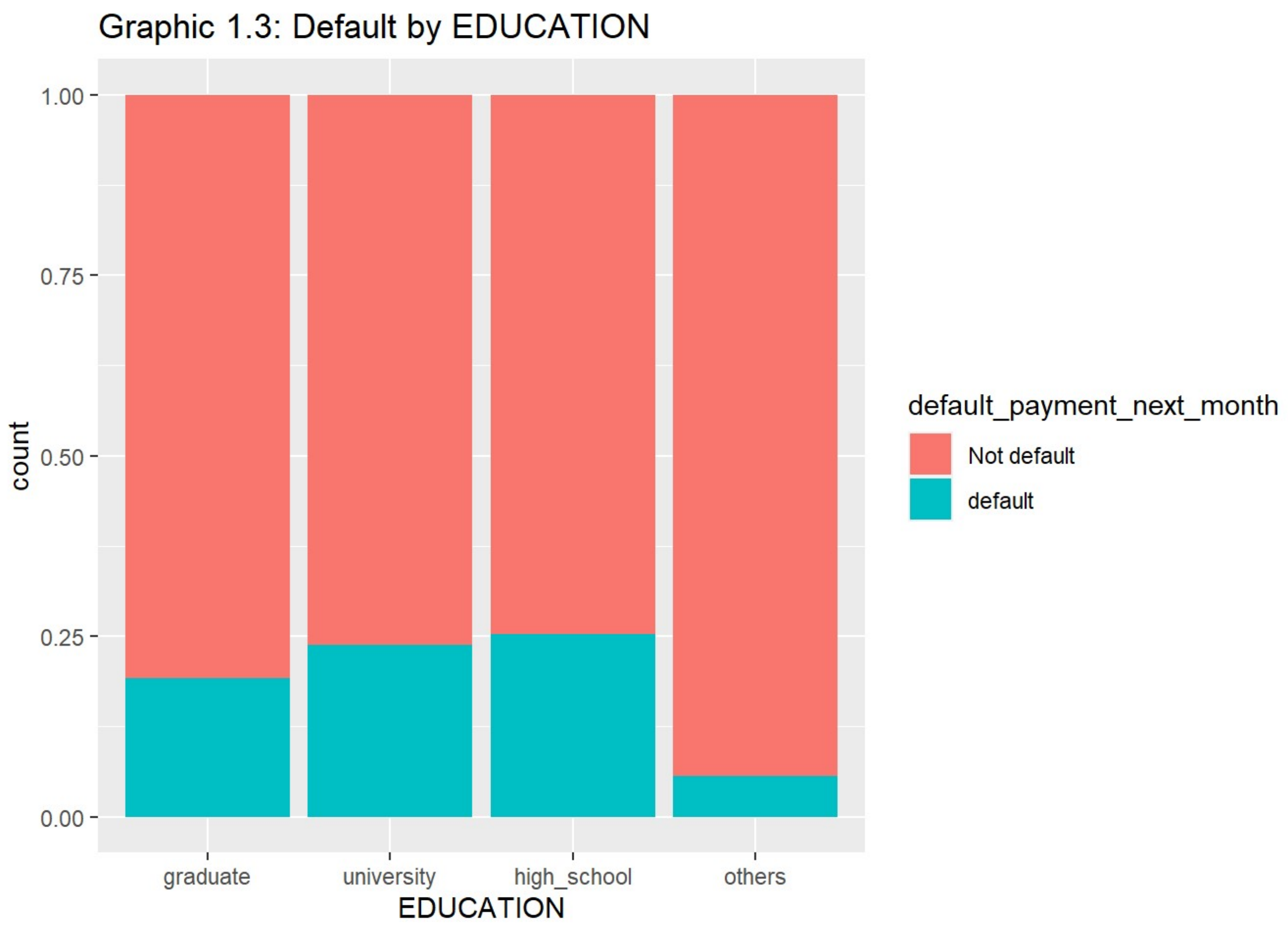
```
ggplot(data = bank, mapping = aes(x = SEX, fill = default_payment_next_month)) + g
eom_bar(position = "fill") + labs(title = "Graphic 1.1: Default by SEX")
```



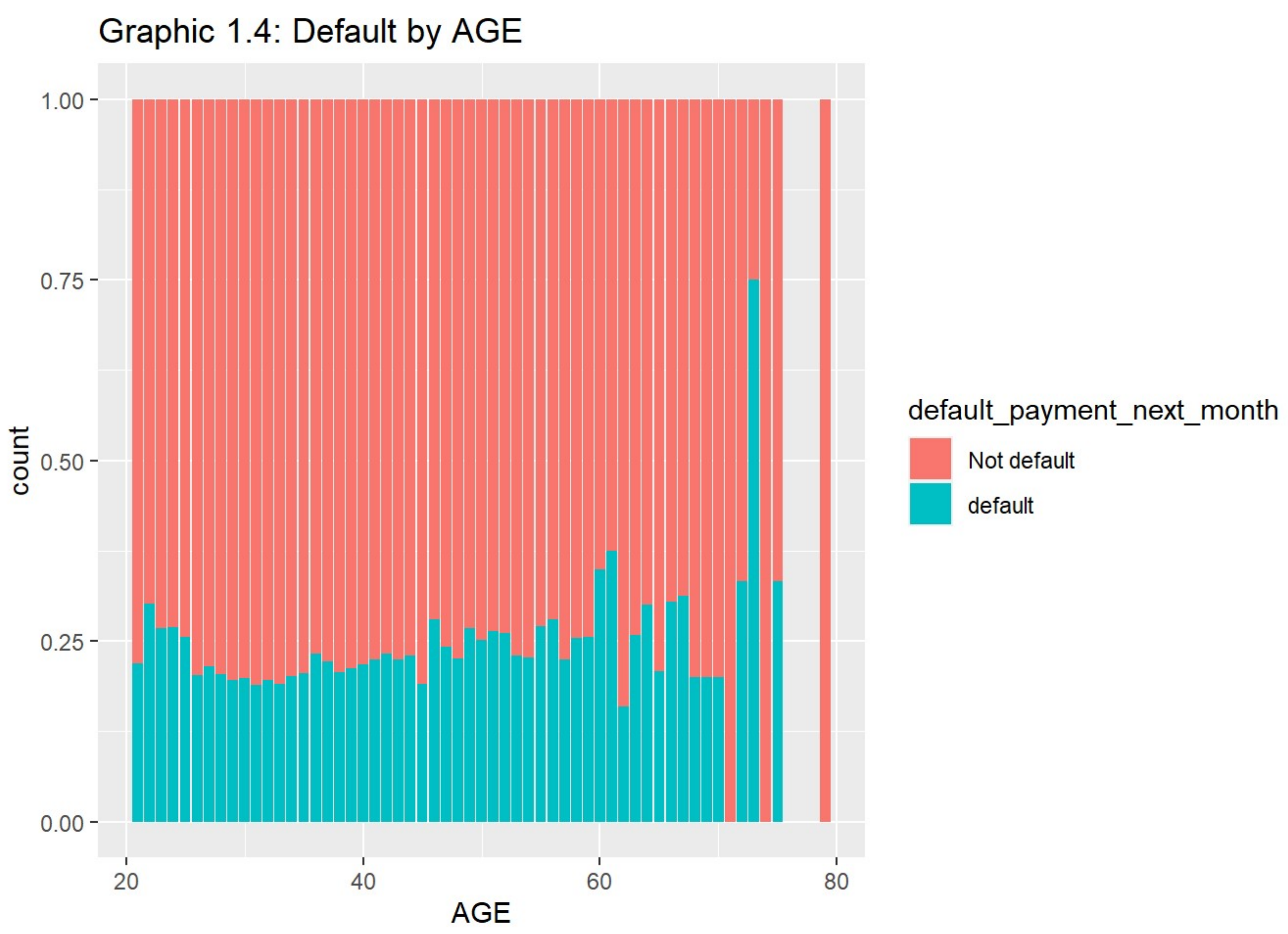
```
ggplot(data = bank, mapping = aes(x = MARRIAGE, fill = default_payment_next_mont
h)) + geom_bar(position = "fill") + labs(title = "Graphic 1.2: Default by MARRIAG
E")
```



```
ggplot(data = bank, mapping = aes(x = EDUCATION, fill = default_payment_next_mont
h)) + geom_bar(position = "fill") + labs(title = "Graphic 1.3: Default by EDUCATIO
N")
```



```
ggplot(data = bank, mapping = aes(x = AGE, fill = default_payment_next_month)) + g
eom_bar(position = "fill") + labs(title = "Graphic 1.4: Default by AGE")
```



Observing the age averages, there is no difference in the averages or in the standard deviations between defaulters and non-defaulters. This is an indication that age does not include being default or not default.

```
bank %>% group_by(default_payment_next_month) %>% summarise(mean_AGE = mean(AGE),
sd_AGE = sd(AGE))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

| default_payment_next_month <fct> | mean_AGE <dbl> | sd_AGE <dbl> |
|---|---|---|
| Not default | 35.39285 | 9.069673 |
| default | 35.71204 | 9.693160 |

2 rows

In the case of previous payments, there is a difference in the averages between detaults and not defaults. The average payout for defaults is less than for non-defaults. Another point is the difference in the standard deviation: the defaults have a smaller standard deviation. This difference may be evidence that the amount of payments is related to the default and not default cases.

```
bank %>% group_by(default_payment_next_month) %>% summarise(mean_PAYAMT1 = mean(PA
Y_AMT1), sd_PAYAMT1 = sd(PAY_AMT1), mean_PAYAMT2 = mean(PAY_AMT2), sd_PAYAMT2 = sd
(PAY_AMT2), mean_PAYAMT3 = mean(PAY_AMT3), sd_PAYAMT3 = sd(PAY_AMT3))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

| default_payment_next_month<br><fct> | mean_PAYA…<br><dbl> | sd_PAYA…<br><dbl> | mean_PAYA…<br><dbl> | sd_PAYA…<br><dbl> | me |
|---|---|---|---|---|---|
| Not default | 6305.575 | 18062.906 | 6616.276 | 25381.19 | |
| default | 3365.575 | 9359.573 | 3382.855 | 11757.61 | |

2 rows

When observing the invoice values, a similarity is noted in the case of detault with the not default. Both averages and standard deviations are similar for both profiles, indicating that the invoice amounts do not determine whether the observations will be defaut and not default.

```
bank %>% group_by(default_payment_next_month) %>% summarise(mean_BILLAMT1 = mean(B
ILL_AMT1), sd_BILL_AMT1 = sd(BILL_AMT1), mean_BILLAMT2 = mean(BILL_AMT2), sd_BILL_
AMT2 = sd(BILL_AMT2), mean_BILLAMT3 = mean(BILL_AMT3), sd_BILL_AMT1 = sd(BILL_AMT
1))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

| default_payment_next_month<br><fct> | mean_BILLA…<br><dbl> | sd_BILL_A…<br><dbl> | mean_BILLA…<br><dbl> | sd_BILL_A.<br><d |
|---|---|---|---|---|
| Not default | 51716.44 | 73277.66 | 49463.30 | 70730 |
| default | 48314.87 | 73636.21 | 47127.88 | 71570 |

2 rows

Below you can see the difference in the amount of the credit according to the default and not default situation. The average of credit_amout and the standard deviation are higher in cases not default.

```
bank %>% group_by(default_payment_next_month) %>% summarise(mean_credit_amount = m
ean(credit_amount), sd_credit_amount = sd(credit_amount))
```

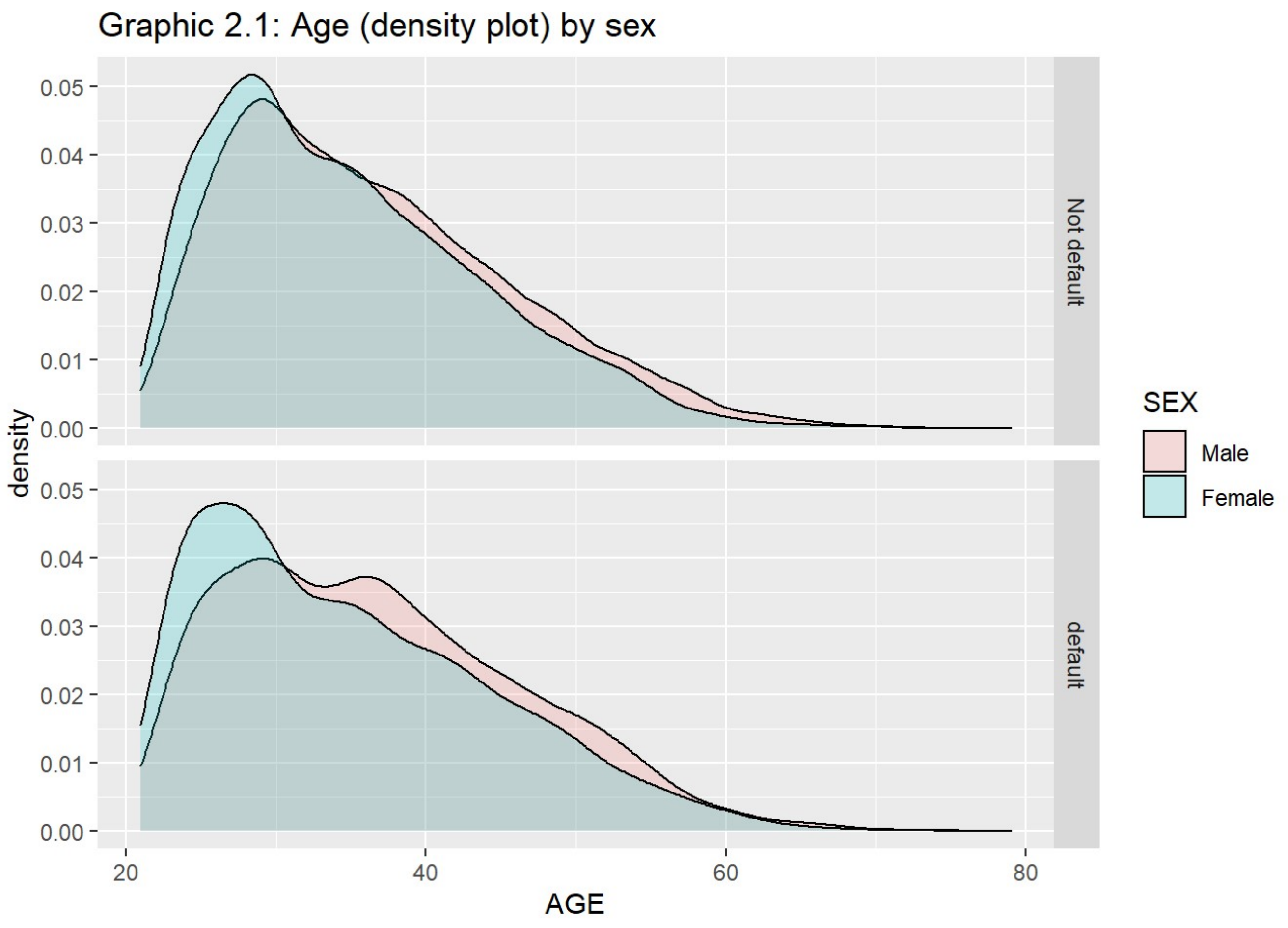
```
## `summarise()` ungrouping output (override with `.groups` argument)
```

| default_payment_next_month<br><fct> | mean_credit_amount<br><dbl> | sd_credit_amount<br><dbl> |
|---|---|---|
| Not default | 178300.0 | 131876.8 |
| default | 130125.3 | 115424.1 |

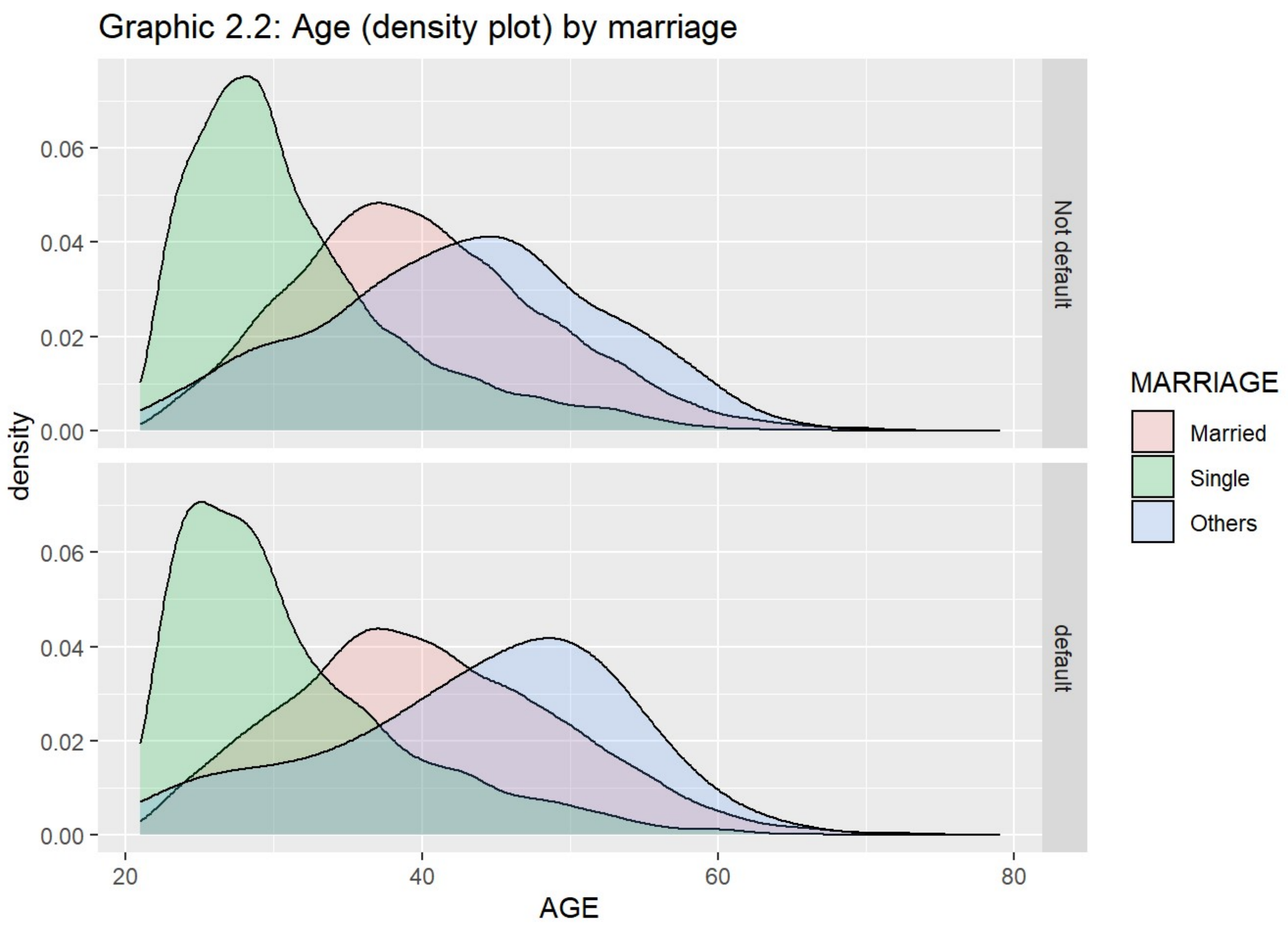
2 rows

In Graphs 2.1, 2.2, 2.3 and 2.4 it will be possible to analyze the distribution of defaulting individuals according to age divided according to educational level, sex and marital status. Graph 2.1 shows that in both women and men, most of the not default are approximately 30 years old. In default, the concentration is well defined in females (20-30 years old) and in males it is more distributed until 40 years old. It is also noted that from 20 to 30 years of age, there are, proportionally, more default women than men. The proportion of men increases after 40 years of age, until approximately 60 years of age, where the proportions are practically equal. Graph 2.2 shows that there is not much difference in the distribution between default and non-default. In the case of single, both default and non-default are concentrated until the age of 30. For the married case, there is a concentration in the beech from 30 to 50 years old. In other cases, the concentration is after 40 years. Graph 2.3 contains the age distribution of default and not default according to the level of education. For both cases (default and not default) the number of observations with university and graduate are concentrated from 20 to 40 years old. In the case of high school, the observations, both default and non-default, are well distributed. For others education, the defaults were concentrated from 20 to 30 years old.

```
ggplot(data = bank, mapping = aes(AGE, fill = SEX)) + geom_density(col = "black",
alpha = 0.2) + labs(title = "Graphic 2.1: Age (density plot) by sex") + facet_grid
(bank$default_payment_next_month)
```



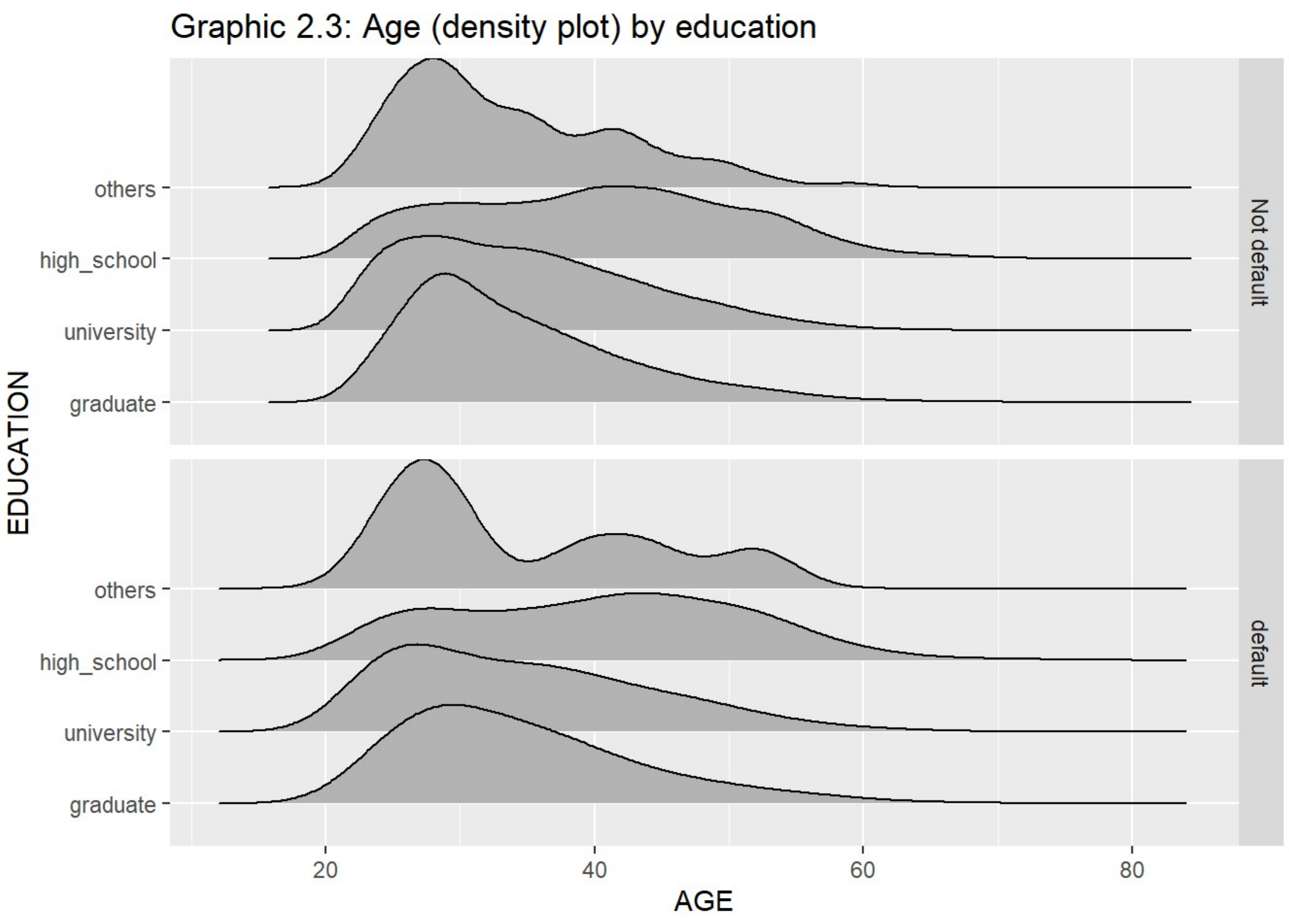
```
ggplot(data = bank, mapping = aes(AGE, fill = MARRIAGE)) + geom_density(col = "bla
ck", alpha = 0.2) + facet_grid(bank$default_payment_next_month) + labs(title = "Gr
aphic 2.2: Age (density plot) by marriage")
```



```
ggplot(data = bank, mapping = aes(AGE, EDUCATION)) + geom_density_ridges() + labs
(title = "Graphic 2.3: Age (density plot) by education") + facet_grid(bank$default
_payment_next_month)
```

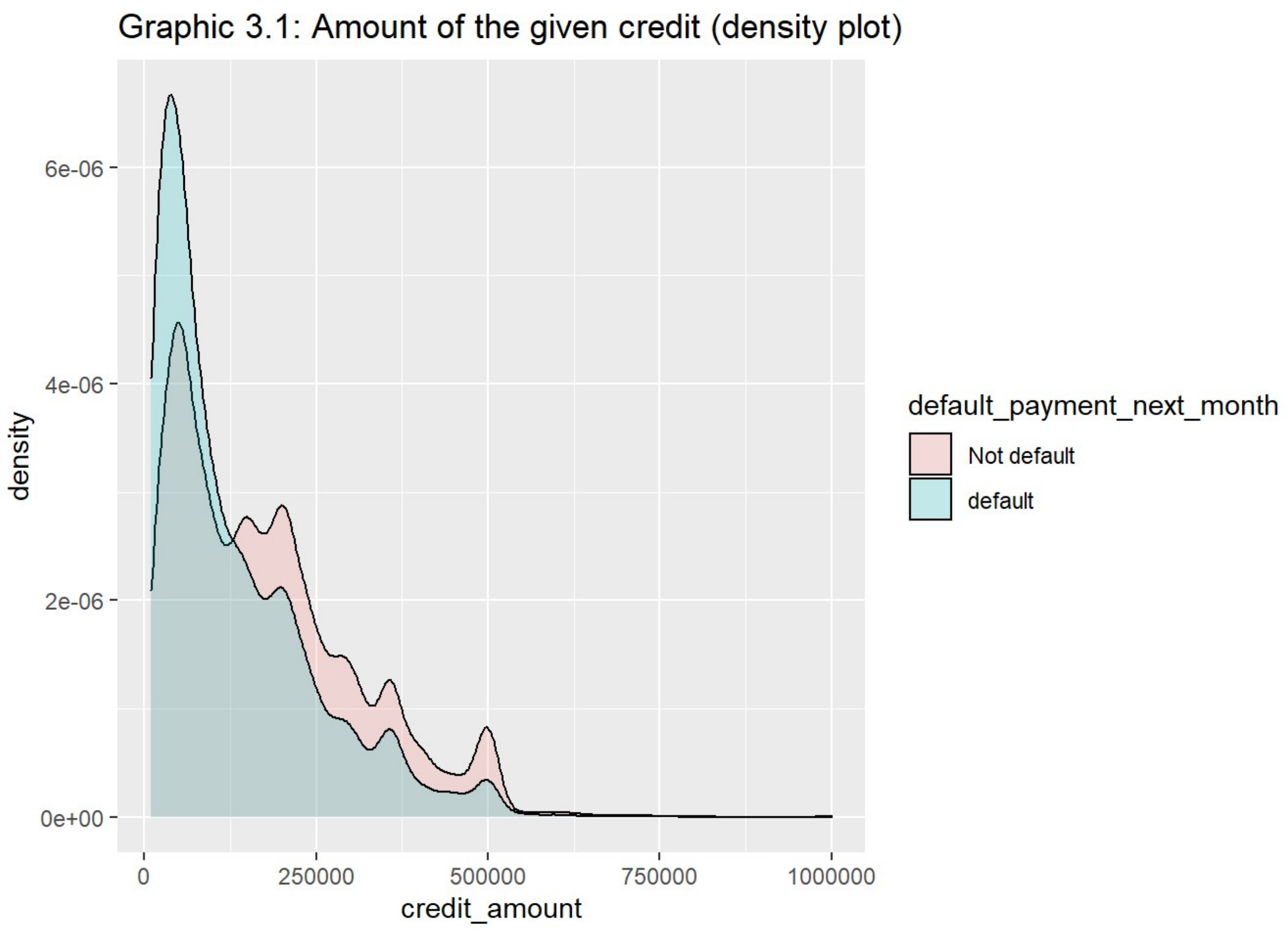
```
## Picking joint bandwidth of 1.77
```

```
## Picking joint bandwidth of 3
```

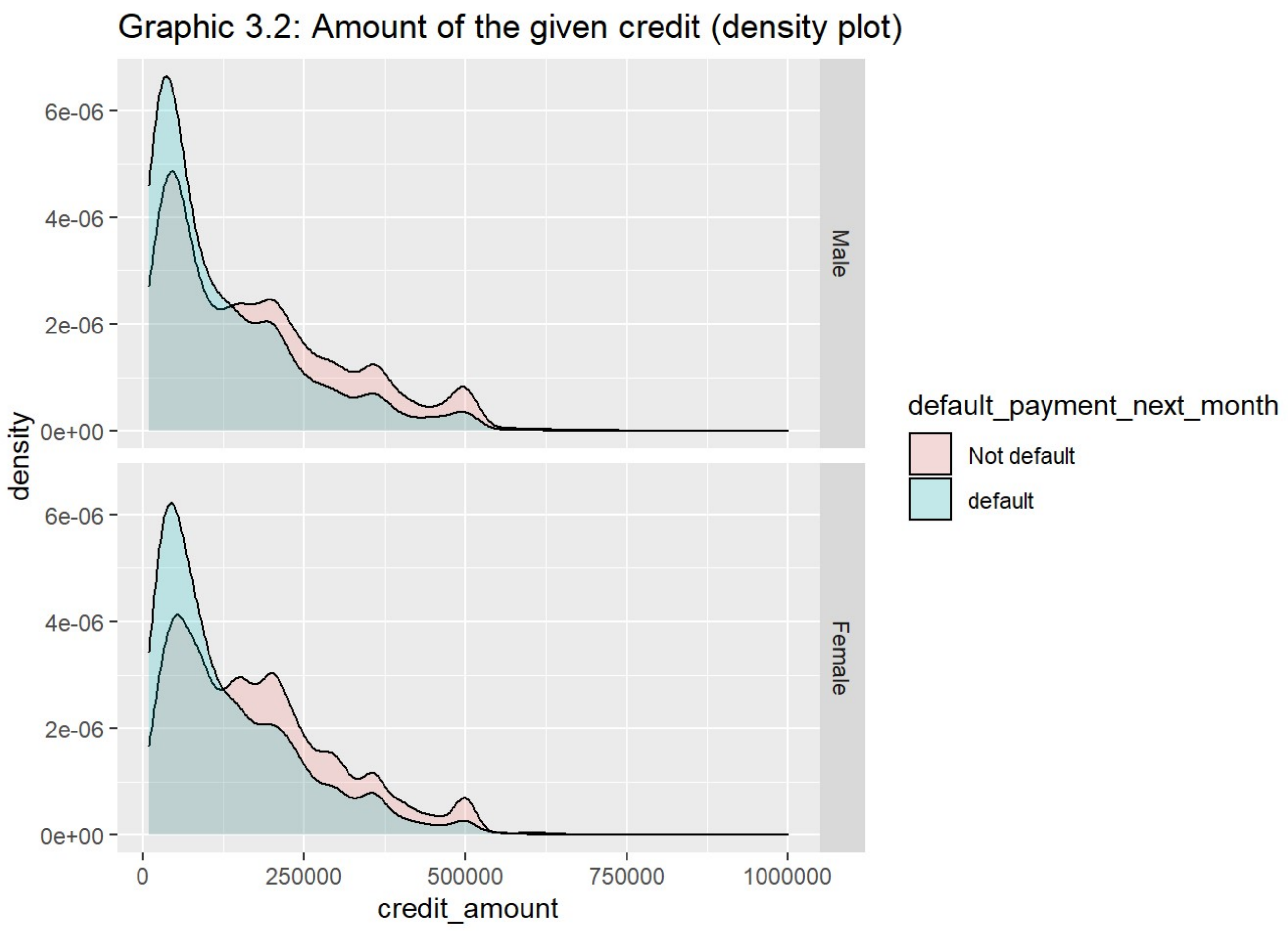


Using Graph 3.1, it can be seen that a large part of the credits assigned both in detault and in non-default are less than $ 125,000. Also note that the higher the value of the assigned credit, the less the proportion of detault. Graph 3.2 shows the distribution of credits assigned according to sex. According to graph 3.2 there is not much difference in the distribution of credits according to sex. In both cases, as in the general case (Graph 3.1), the default cases are concentrated in the smallest amounts assigned. Graph 3.3 shows the distribution according to marital status. Once again, it is the same scenario as the general case: most of the default cases are concentrated in the lower amounts of credit assigned. Finally, in Graph 3.4 there is the distribution of credits assigned according to education. Looking at graph 3.4, it is noted that for cases in which education is graduate, the volume of credit assigned is well distributed, which is not the case for other education. Another point to highlight is the default credits for other schooling. In this case, default credits are concentrated in the lower amounts, in contrast to those in not default, which have mostly higher values than those in default.

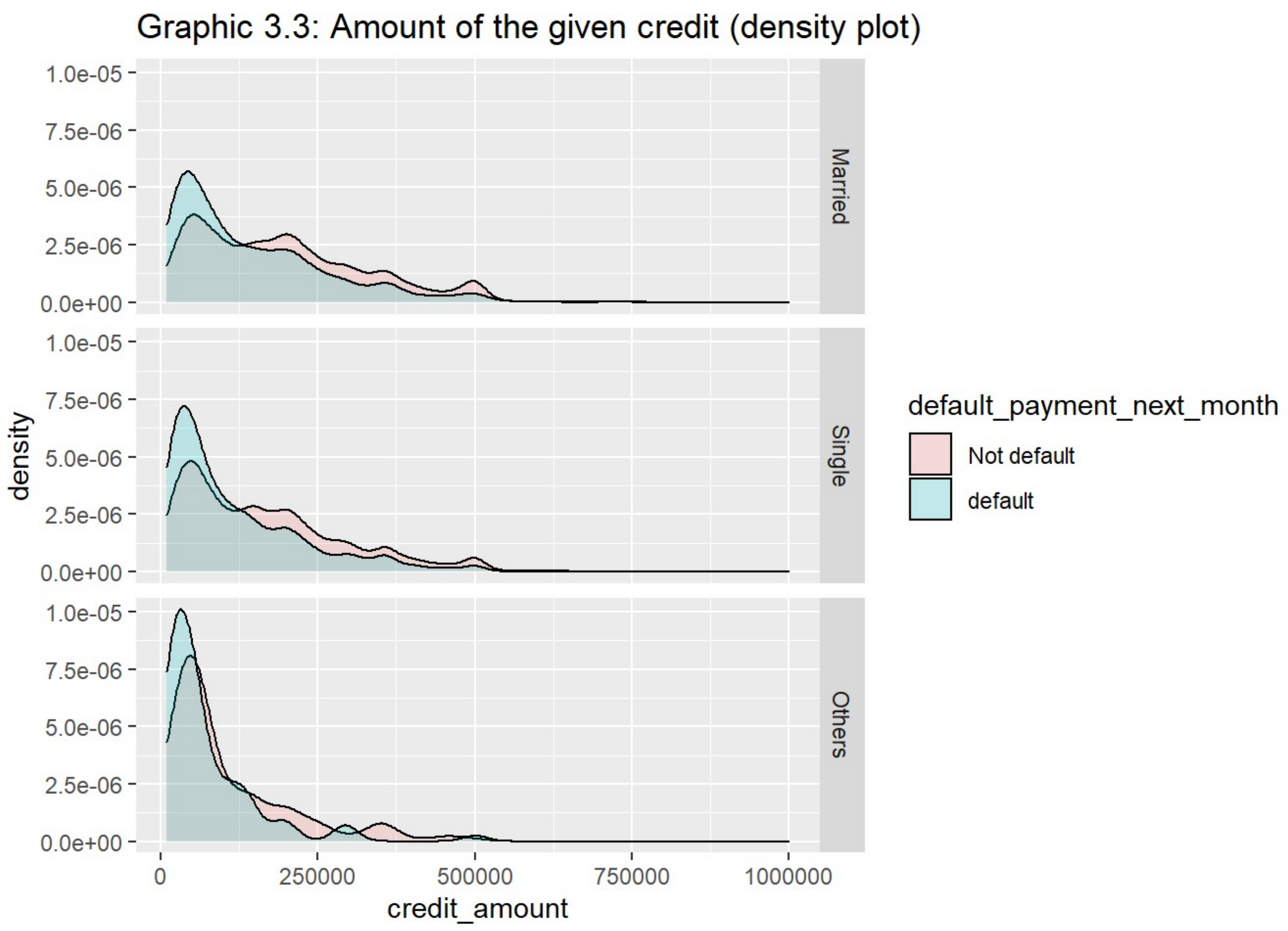
```
ggplot(data = bank, mapping = aes(credit_amount, fill = default_payment_next_mont
h)) + geom_density(col = "black", alpha = 0.2) + labs(title = "Graphic 3.1: Amount
of the given credit (density plot)")
```



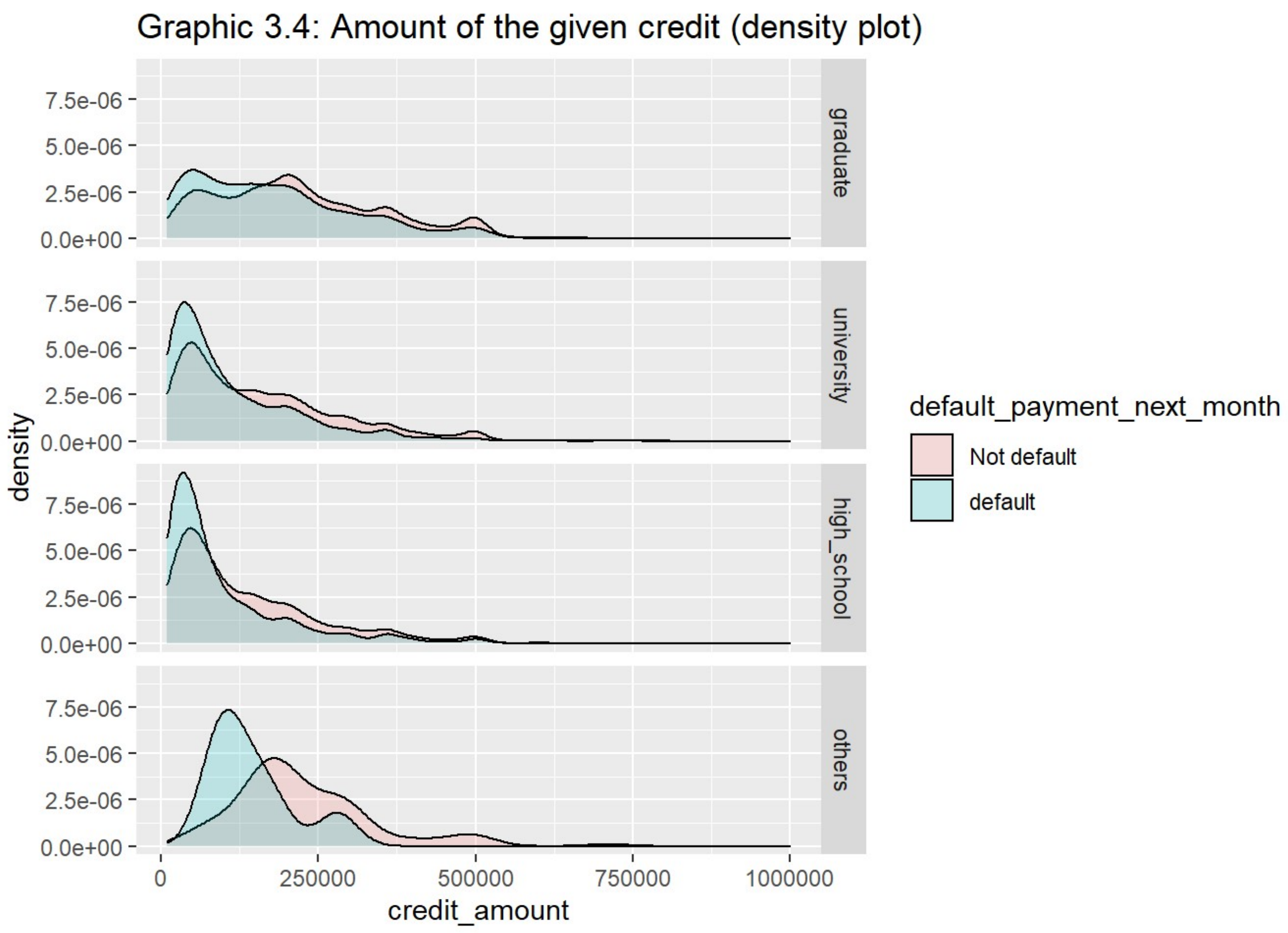
```
ggplot(data = bank, mapping = aes(credit_amount, fill = default_payment_next_mont
h)) + geom_density(col = "black", alpha = 0.2) + facet_grid(bank$SEX) + labs(title
= "Graphic 3.2: Amount of the given credit (density plot)")
```



```
ggplot(data = bank, mapping = aes(credit_amount, fill = default_payment_next_mont
h)) + geom_density(col = "black", alpha = 0.2) + facet_grid(bank$MARRIAGE) + labs
(title = "Graphic 3.3: Amount of the given credit (density plot)")
```



```
ggplot(data = bank, mapping = aes(credit_amount, fill = default_payment_next_mont
h)) + geom_density(col = "black", alpha = 0.2) + facet_grid(bank$EDUCATION) + labs
(title = "Graphic 3.4: Amount of the given credit (density plot)")
```



# Methods

The models adopted were logistic and knn, because the explained variable is binary or classification (default or not default).
The variables were selected according to the analysis in the previous section. The variables chosen were: credit_amount, PAYAMT_0, PAYAMT_1, PAYAMT_2, MARRIAGE and EDUCATION.
The function `createDataPartition` was used to create a partition to train the algorithm (60% of the database) and one to test the algorithm (40%). This separation is important to prevent the same database, or similar databases, from being used to train the algorithm and to calculate the values, that is, to avoid overtraining.
To evaluate performance, F_1was used as a parameter. It is important to take specificity into account, because more important than identifying cases of non-default is to identify cases of default, as it does not foresee these cases which can cause damage to the bank. Another precaution to be adopted was the prevalence value, which cannot be close to 1 or 0.

```
index <- createDataPartition(bank$default_payment_next_month, p = 0.6, list = FALS
E)
test_set <- bank[-index,]
train_set <- bank[index,]
```

```
## Warning: The `i` argument of ``[`()` can't be a matrix as of tibble 3.0.0.
## Convert to a vector.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

## Logistic regression

Logitic is a method that calculates used for binary categorical data. Similar to the case of heads or tails. Either it’s expensive, or it’s a crown. In the case of this work: it is either default or not default. The result of the logit function is a value that varies from 0 to 1 and can be interpreted as a probability.
After calculating the estimated values, the next step was to decide what values of x> p can be considered default. For this, simulations were performed for all valuesof x between the lowest p value and the highest p value obtained in the estimation. For each simulation, F_1 was calculated using the `F_meas` function. The value of x with the highest F1 was used as a parameter to consider default (if p> x) not default (p <x).
Finally, two parameters will be evaluated: accuracy and, mainly, specificity. Specificity is important because the most important thing is to identify defaults when they are default. After choosing the best cut, all observations that obtained values greater than the cut value were considered default. Contrary cases were considered not default.

###knn The other algorithm is performed using knn as it is easy to adapt to multiple dimensions. After estimating by the `knn3` function. To obtain the best k, the estimation for k ranging from 1 to 100 was performed. High values for k were not adopted, since large values of k cause similarity to a linear regression, not allowing flexibility. For each simulation, F_1 was calculated using the `F_meas` function to evaluate the best k.
After performing the estimation with the best k obtained, acccuracy, specificity and prevalence were observed. It was important to observe the specificity, because the important thing is to identify which observations will be default.
#Results ##logistic Observing the results obtained, it is noted that the lowest estimated value was 0.012 and the highest was 0.9897. The choice of the best cutoff (ranging from 0.012 to 0.098 with an increase of 0.01) was 0.452564. This means that if the calculated value is greater than 0.262564, the observation was considered to be default, otherwise it was considered non-default. After applying this criterion to the calculated values, the Accuracy was approximately 80%, that is, in 80% of the cases the algorithm will correct whether or not the observation is default. Another important observed value was Specificity. For cases in which the observation is default, the algorithm will correct approximately 0.48% of the cases. The prevalence value was 0.77, not considered close to 1.

```
fit1 <- glm(default_payment_next_month ~ credit_amount + PAY_0 + PAY_2 + PAY_3 + E
DUCATION + MARRIAGE, data = train_set, family = "binomial")
fit1
```

```
##
## Call:  glm(formula = default_payment_next_month ~ credit_amount + PAY_0 +
##     PAY_2 + PAY_3 + EDUCATION + MARRIAGE, family = "binomial",
##     data = train_set)
##
## Coefficients:
##          (Intercept)         credit_amount                 PAY_0
##           -9.534e-01            -1.767e-06             6.347e-01
##                PAY_2                 PAY_3   EDUCATIONuniversity
##            5.573e-02             9.545e-02            -8.304e-02
## EDUCATIONhigh_school       EDUCATIONothers        MARRIAGESingle
##           -5.317e-02            -7.687e-01            -1.961e-01
##       MARRIAGEOthers
##           -1.482e-01
##
## Degrees of Freedom: 17760 Total (i.e. Null);  17751 Residual
## Null Deviance:       18860
## Residual Deviance: 16660     AIC: 16680
```

```
pred1 <- predict(fit1, test_set, type = "response")

min(pred1)
```

```
## [1] 0.01244842
```

```
max(pred1)
```

```
## [1] 0.9930957
```

```
cutoff <- seq(min(pred1),max(pred1), 0.01)
acc <- map_dbl(cutoff, function(x){
  pred1 <- ifelse(pred1 > x, 1, 0)
  pred1 <- factor(pred1, levels = c(0,1), labels = c("Not default","default"))
  confusionMatrix(pred1, test_set$default_payment_next_month)$byClass["Balanced Ac
curacy"]
})
acc
```

```
##  [1] 0.5000544 0.5002456 0.5018704 0.5089272 0.5138639 0.5226118 0.5298316
##  [8] 0.5363790 0.5447464 0.5513583 0.5569957 0.5589185 0.5654236 0.5717072
## [15] 0.5763077 0.5841093 0.5954001 0.6091149 0.6232686 0.6403195 0.6524439
## [22] 0.6668635 0.6767591 0.6842991 0.6874501 0.6857570 0.6857792 0.6834216
## [29] 0.6807641 0.6794837 0.6782515 0.6765322 0.6760812 0.6748169 0.6733391
## [36] 0.6724070 0.6697937 0.6674261 0.6655960 0.6629264 0.6583421 0.6503533
## [43] 0.6451168 0.6409311 0.6370958 0.6346456 0.6286258 0.6237375 0.6177399
## [50] 0.6111745 0.6058010 0.6002363 0.5939689 0.5873773 0.5777054 0.5663302
## [57] 0.5611460 0.5565013 0.5501252 0.5430444 0.5383433 0.5311820 0.5271836
## [64] 0.5261307 0.5256173 0.5251039 0.5244012 0.5231308 0.5211034 0.5199679
## [71] 0.5193196 0.5161044 0.5136985 0.5114275 0.5102658 0.5077250 0.5074009
## [78] 0.5066439 0.5066439 0.5066439 0.5062654 0.5056976 0.5052386 0.5043205
## [85] 0.5043749 0.5035917 0.5032132 0.5024562 0.5021321 0.5021864 0.5022408
## [92] 0.5016469 0.5016469 0.5017012 0.5015120 0.5012966 0.5011073 0.5005396
## [99] 0.5001893
```

```
best <- cutoff[which.max(acc)]
best
```

```
## [1] 0.2524484
```

```
pred1 <- ifelse(pred1 > best, 1, 0)
pred1 <- factor(pred1, levels = c(0,1), labels = c("Not default","default"))
confusionMatrix(pred1, test_set$default_payment_next_month)
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    Not default default
##   Not default        7828    1258
##   default            1370    1384
##
##                Accuracy : 0.778
##                  95% CI : (0.7704, 0.7855)
##     No Information Rate : 0.7769
##     P-Value [Acc > NIR] : 0.38357
##
##                   Kappa : 0.3693
##
##  Mcnemar's Test P-Value : 0.03037
##
##             Sensitivity : 0.8511
##             Specificity : 0.5238
##          Pos Pred Value : 0.8615
##          Neg Pred Value : 0.5025
##              Prevalence : 0.7769
##          Detection Rate : 0.6611
##    Detection Prevalence : 0.7674
##       Balanced Accuracy : 0.6875
##
##        'Positive' Class : Not default
##
```

##knn After running the algorithm, the best k, considering the old specifitiy value, was k = 11. With k = 2 the cutoff that obtained the highest specificity. Using k = 11, accuracy greater than 80% and specififity of 0.33% were obtained, that is, the algorithm is able to correct 80% of cases if the observation is default or not default. If the observation is default, the algorithm hits 34% of the cases. The prevalence value was similar to the logistic, approximately 0.77.

```
ks <- seq(1,100,by = 1)
overall_acc_k <- map_df(ks, function(k){
fit2 <- knn3(default_payment_next_month ~ credit_amount + PAY_0 + PAY_2 + PAY_3 +
EDUCATION + MARRIAGE, k = k, data = train_set)
pred2 <- predict(fit2, test_set, type = "class")
tibble(confusionMatrix(pred2, test_set$default_payment_next_month)$byClass["Specif
icity"], k)
})
overall_acc_k
```

| confusionMatrix(pred2, test_set$default_payment_next_month)$byClass["S |
|---|
| |
| |
| |

| confusionMatrix(pred2, test_set$default_payment_next_month)$byClass["S |
|---|
| |
| |
| |
| |
| |
| |

1-10 of 100 rows Previous **1** 2 3 4 5 6 ... 10 Next

```
max_acc <- which.max(overall_acc_k$`confusionMatrix(pred2, test_set$default_paymen
t_next_month)$byClass["Specificity"]`)
max_acc
```

```
## Specificity
##           6
```

```
fit_max<- knn3(default_payment_next_month ~ credit_amount + PAY_0 + PAY_2 + PAY_3
+ EDUCATION + MARRIAGE, k = max_acc, data = train_set)
predmax <- predict(fit_max, test_set, type = "class")
confusionMatrix(predmax, test_set$default_payment_next_month)
```

```
## Confusion Matrix and Statistics
##
##               Reference
## Prediction    Not default default
##   Not default        8639    1727
##   default             559     915
##
##                Accuracy : 0.8069
##                  95% CI : (0.7997, 0.814)
##     No Information Rate : 0.7769
##     P-Value [Acc > NIR] : 7.355e-16
##
##                   Kappa : 0.339
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9392
##             Specificity : 0.3463
##          Pos Pred Value : 0.8334
##          Neg Pred Value : 0.6208
##              Prevalence : 0.7769
##          Detection Rate : 0.7296
##    Detection Prevalence : 0.8755
##       Balanced Accuracy : 0.6428
##
##        'Positive' Class : Not default
##
```

## Conclusions and limitations

This work used a database related to bank debts available on the UCL website. This database consists of 29601 observations and 15 variables. When analyzing the database, it was possible to observe that it is composed marjorially by observations of the female sex, single, and that the value of the assigned credit varies from 10000 dollars to 1000000 dollars. It is also concluded that the proportion of people with high school education is the one with the highest proportion of default people. Regarding age, the mean and standard deviation of not default with default are similar, indicating that age does not have much influence on whether or not it is default. On the other hand, the average payment for next month is higher for not default. The methods used were logistic and knn and the criterion adopted to evaluate the best cutoff was F_1. For the case of the ligistic, the algorithm was able to hit 80% of the cases being able to hit 40% of the cases in which the observation was default. The algorithm was considered useful, as the prevalence value was not close to 0 or 1. For Knn's case, the algorithm behaved in a similar way to logistic: 0.8 accurately, 0.33 specificity and 0.77 prevalence. The work was important in developing an algorithm capable of identifying possible default problems. Identifying before it occurs is important for the bank to prevent and plan the reaction to possible default cases. This study found limitations due to the absence of relevant information such as income and equity history.