Factors Affecting Credit Card Default

Introduction

Credit card issuers keep running through an issue where they give credit card to customer and customer start to have default payments. Default payments are the card payments that customer failed to pay. In this project, I will be trying to help credit card issuers figure out the customer that they should give their credit cards to. the data set that I will be using to explain the customer that credit card issuers should focus on were obtained from Kaggle website. Using the data set, I will be explain the relationship between default payment and customer characteristics such as age, education, marital status, repayment status and amount of credit given.

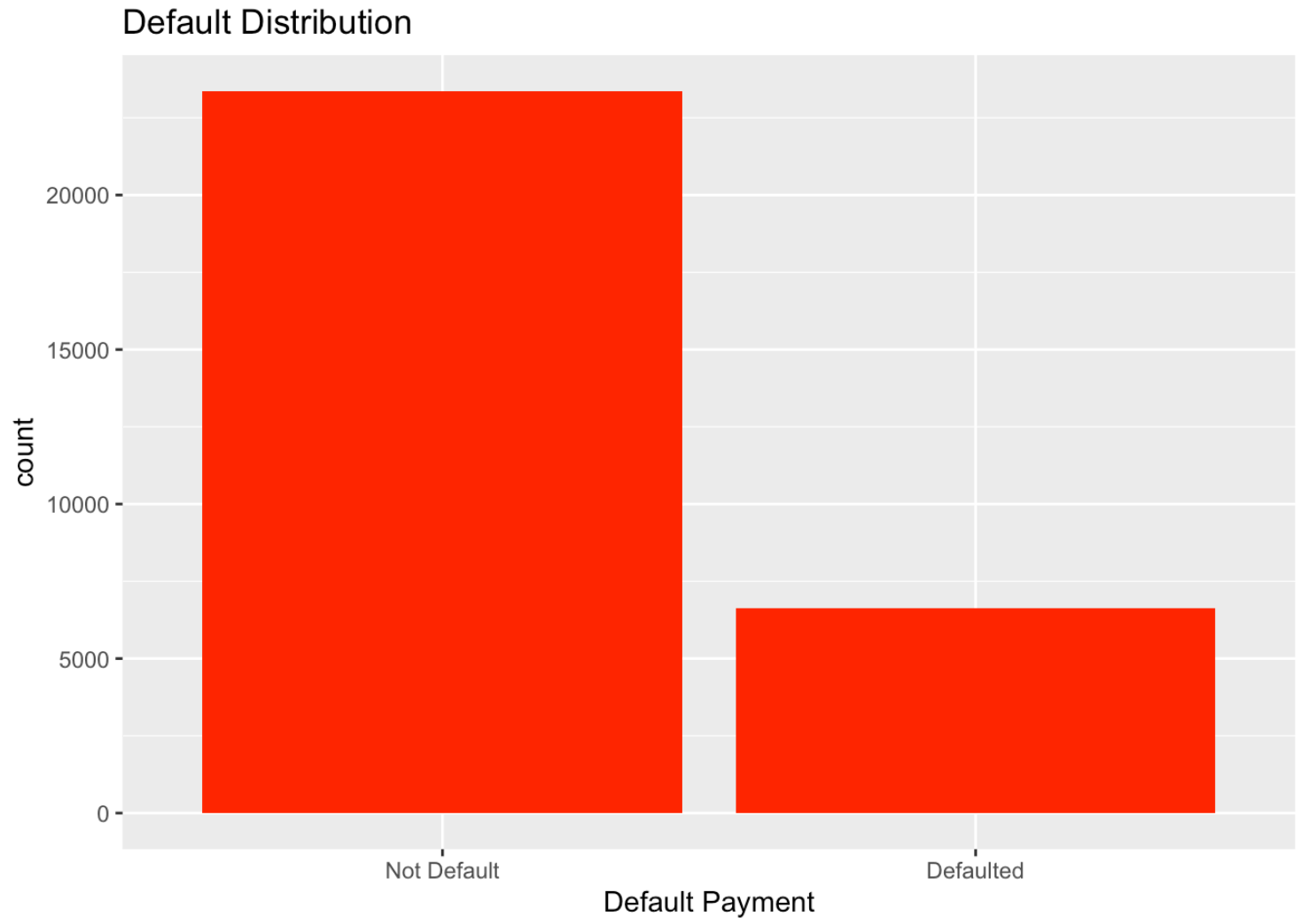
In this report, here are the customer characterisitics that we will use.

1.Age
2.Education
3.Marriage
4.Limited balance 5.Bill Amount 6. Payment Status 1st month 7. Payment Status 2nd month 8. Payment Status 3rd month

```
data <- read.csv("UCI_Credit_Card.csv")
library(ggplot2)
library(ggthemes)
library(dplyr)
library(class)
factor_vars <- c('LIMIT_BAL','EDUCATION','MARRIAGE','AGE','default.payment.next.month','PAY_0','PAY_2','PAY_3')
data[factor_vars] <- lapply(data[factor_vars], function(x) as.factor(x))
A = data
```

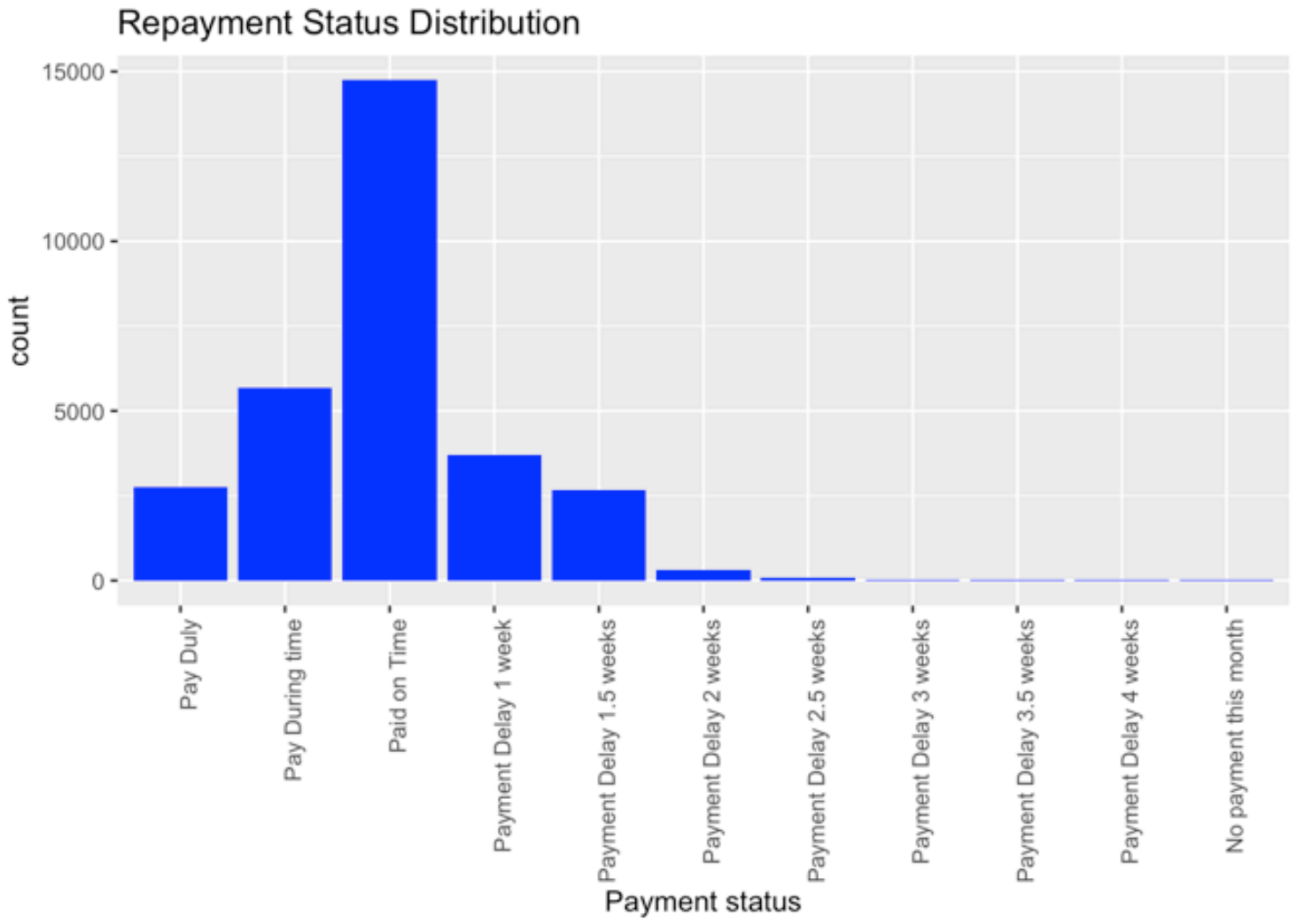
Distributions

```
def <- c("Not Default","Defaulted")
ggplot(data, aes(x=default.payment.next.month))+
  geom_bar(fill="red")+
  labs(y="count",x="Default Payment")+
  ggtitle("Default Distribution")+scale_x_discrete(labels= def)
```



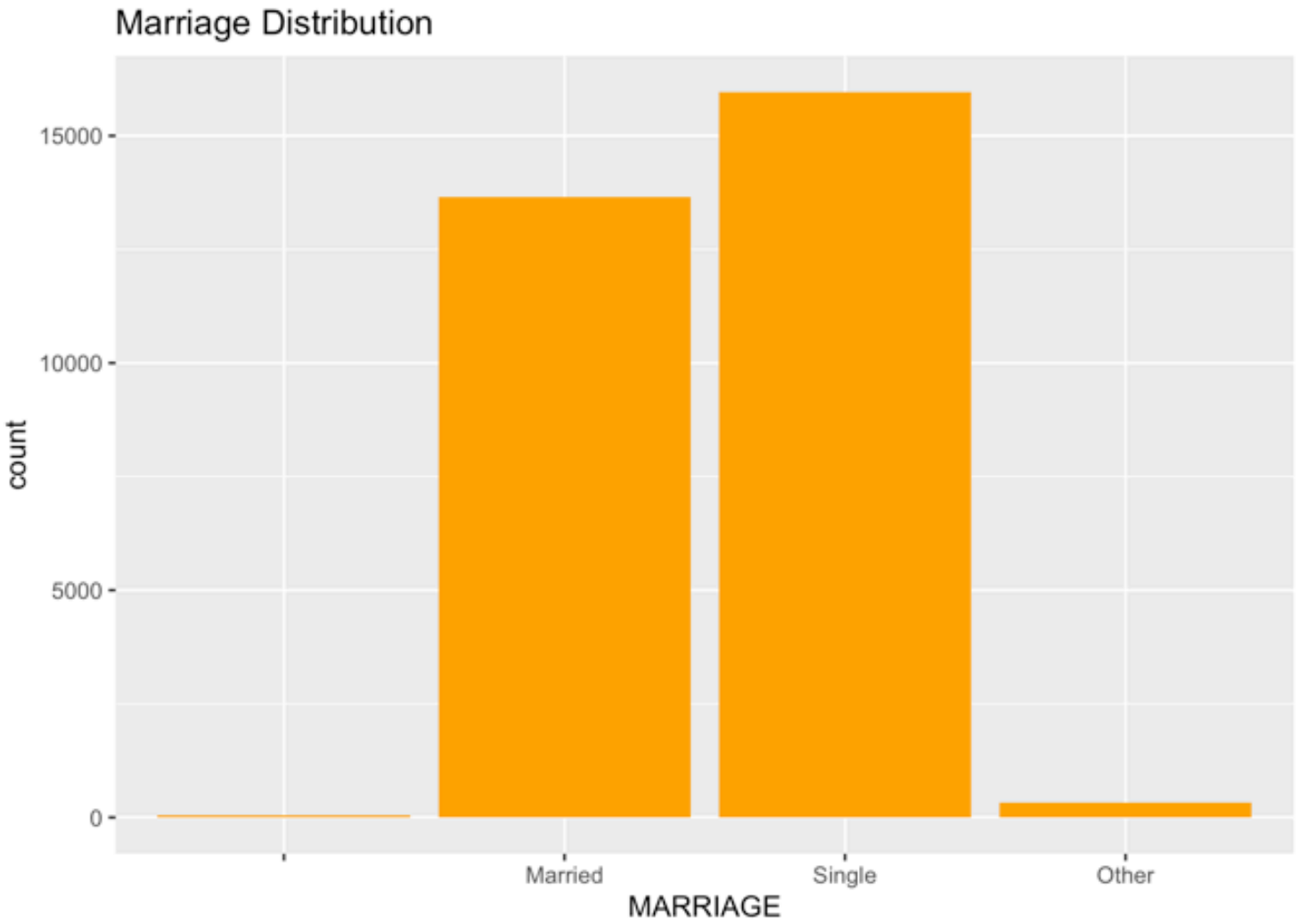
From this graph, we can see that there is a huge amount of people that didnt get credit default. As the number of people that got defaulted is closer 1/3 of the number of people that didnt get credit default.

```
repa <- c("Pay Duly", "Pay During time", "Paid on Time", "Payment De
lay 1 week","Payment Delay 1.5 weeks","Payment Delay 2 weeks","Payme
nt Delay 2.5 weeks","Payment Delay 3 weeks","Payment Delay 3.5 weeks
","Payment Delay 4 weeks","No payment this month")
ggplot(data, aes(x=PAY_0))+
  geom_bar(fill="blue")+
  labs(y="count", x="Payment status")+
  ggtitle("Repayment Status Distribution")+scale_x_discrete(labels=r
epa)+ theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



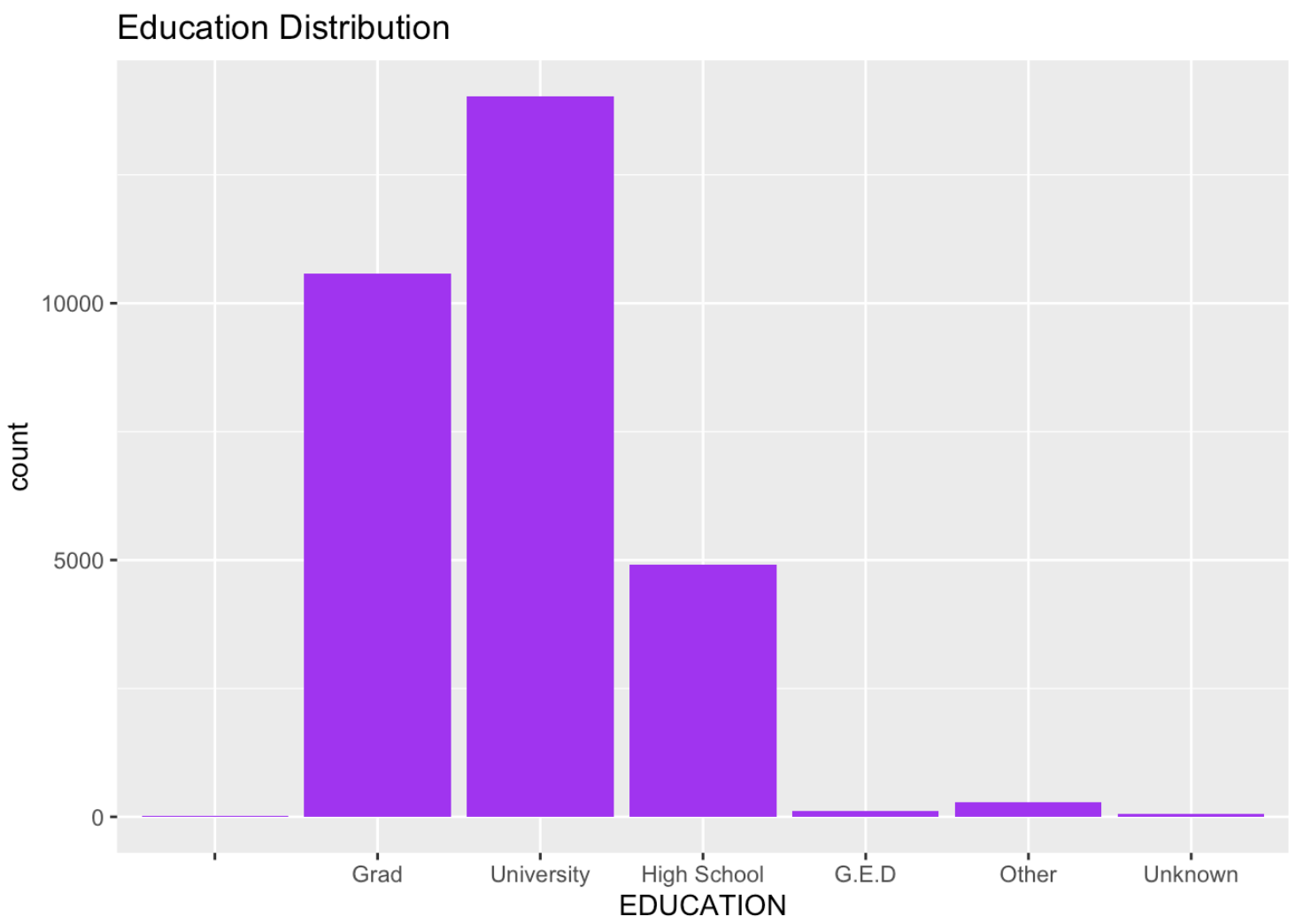
In this graph, we can see normal distribution. Also, we can see that as the payment due date comes, the number of people that make payment increases. As the months of delayed payment increases, the number of people that make payment decreases. This graph is skewed right. This happened due to the fact that many client made their payment on time as only few client had a huge delay in their payment.

```
Mar <- c("","Married","Single","Other")
ggplot(data, aes(x=MARRIAGE))+
  geom_bar(fill="orange")+
  labs(y="count")+
  ggtitle("Marriage Distribution")+ scale_x_discrete(labels= Mar)
```



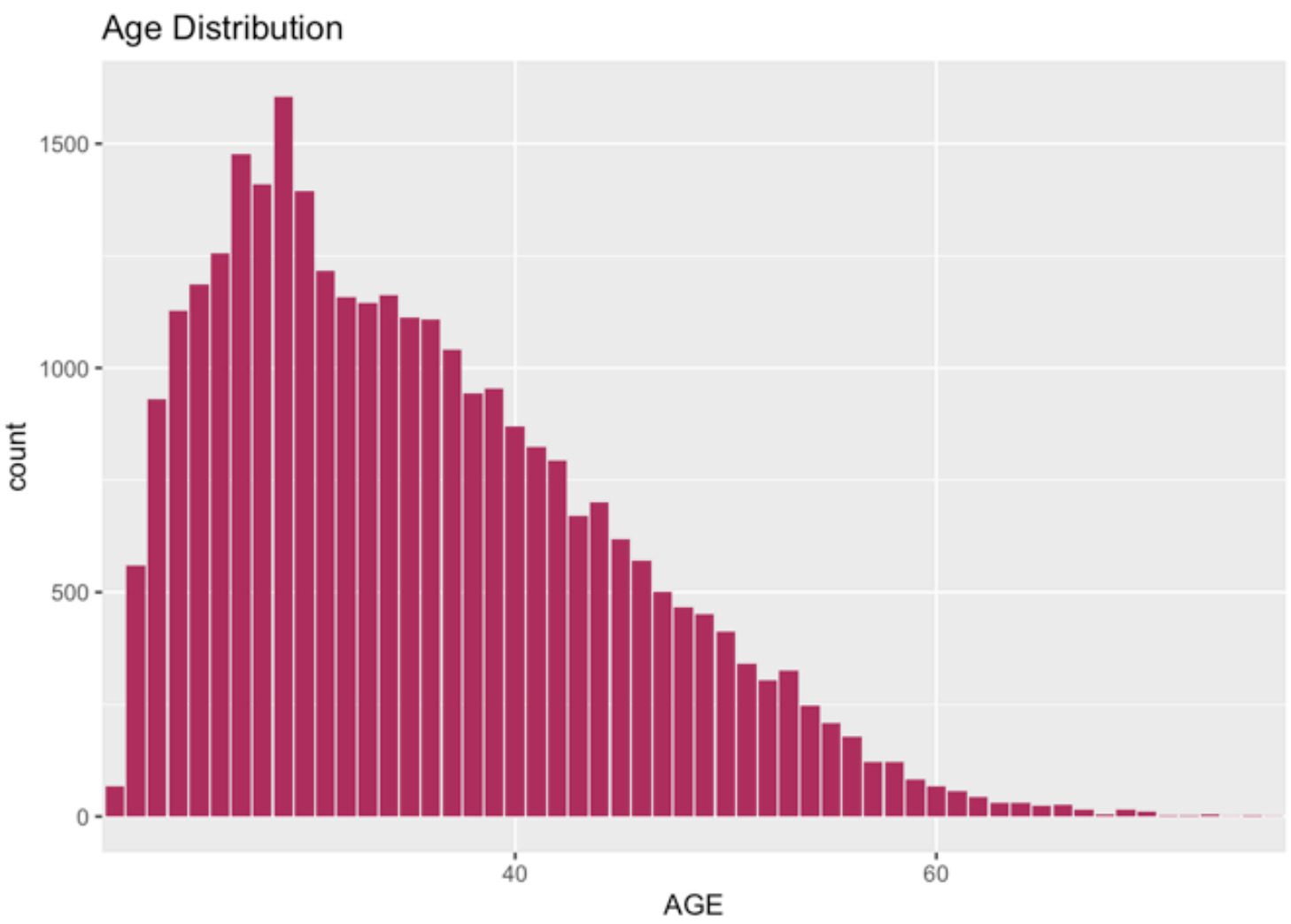
The number of people that are single and apply for credit are the highest. The amount of people that are married get to have the second highest number of people that get credit. Only a few amount of people that mentioned their marital status as other apply for credit.

```
ed <- c("","Grad","University", "High School","G.E.D","Other","Unkno
wn")
ggplot(data, aes(x=EDUCATION))+
  geom_bar(fill="purple")+
  labs(y="count")+
  ggtitle("Education Distribution")+ scale_x_discrete(labels= ed)
```



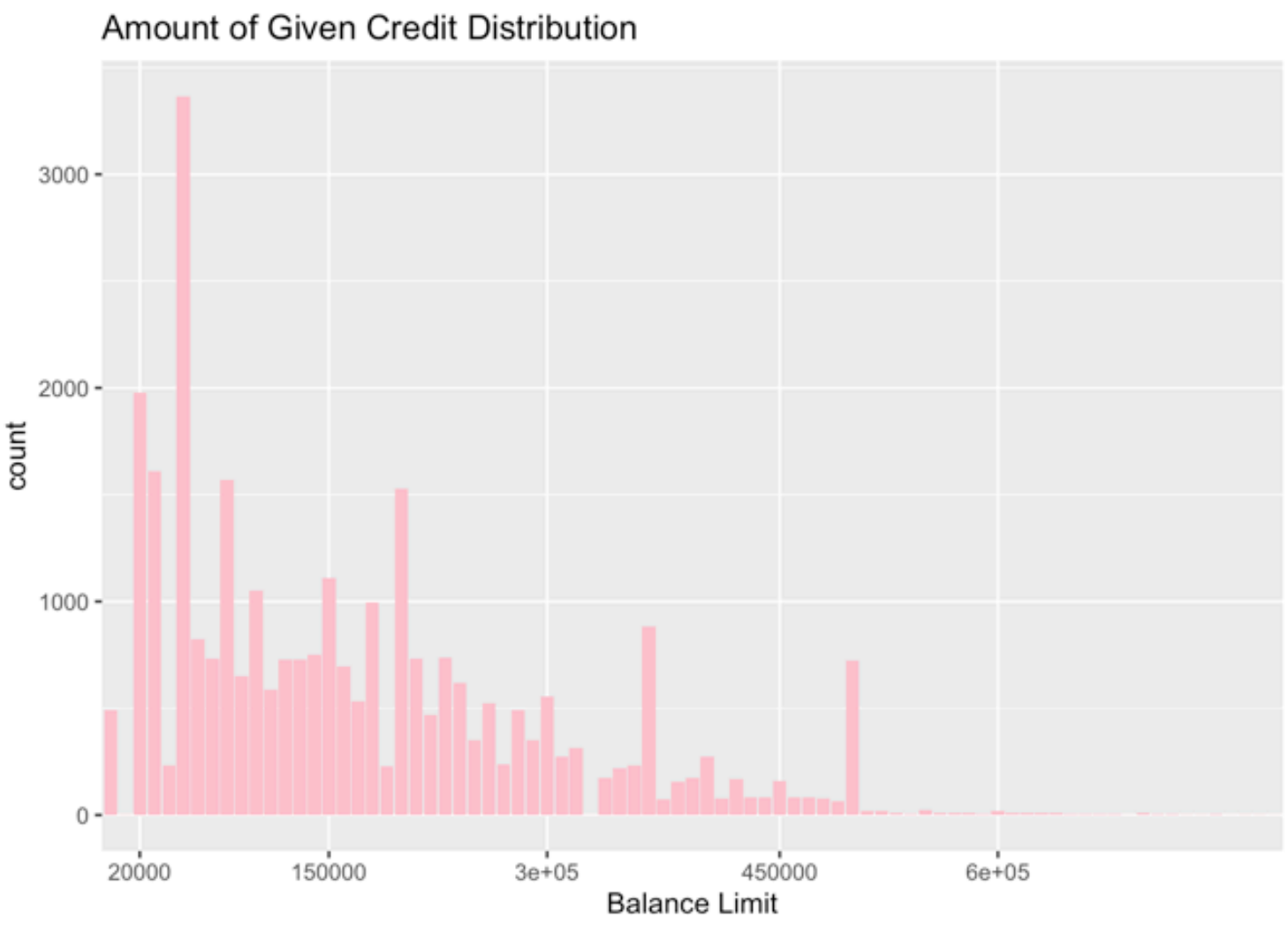
In this graph, we can see that the number of students that are in unversity and apply for credit card is high. The amount of high school students that apply for credit card is about 1/3 the amount of unversity students that apply for card. The number of people that have G.E.D or unsure about their education and applying for credit get to be super low.

```
ggplot(data, aes(x=AGE))+
  geom_bar(fill="maroon")+
  labs(y="count")+scale_x_discrete( breaks=c( 20,40,60,80,100),)+
  ggtitle("Age Distribution")
```



From this graph we can see the age distribution. As we can see, the graph is skewed to the right. The skewness tell us that young client apply the most for credit card. Also, the graph tells us that the number of people exactly between the age of 20-40 that apply for credit card is super high.

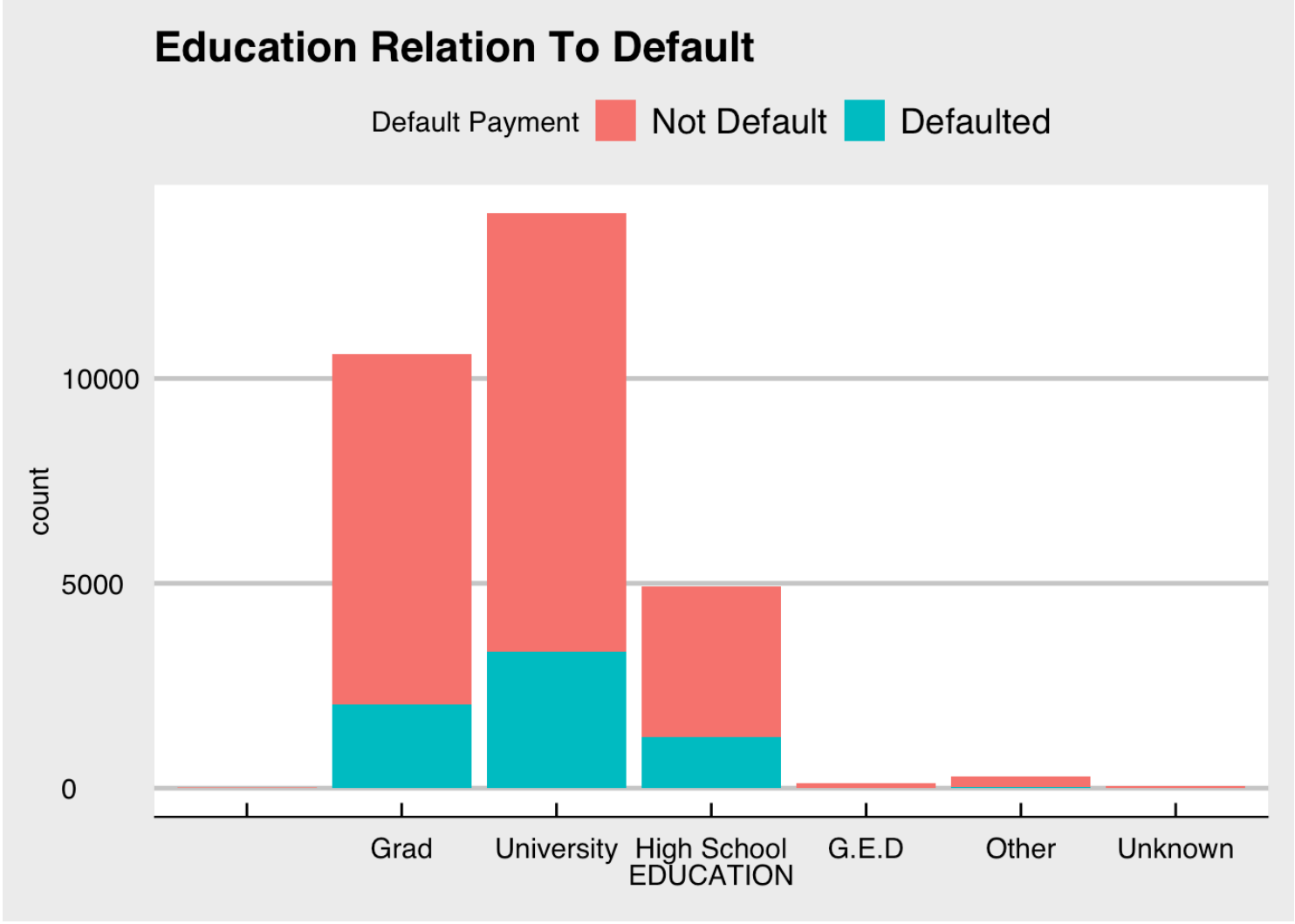
```
ggplot(data, aes(x=LIMIT_BAL))+
  geom_bar(fill="pink")+
  labs(y="count",x="Balance Limit")+
  ggtitle("Amount of Given Credit Distribution")+
  scale_x_discrete( breaks=c( 20000, 150000, 300000,450000,600000),)
```



From this graph, we can determine that the higher number of people have credit at 20,000 and 60,000. Only a few amount of people that get a credit between 450,000 and 600,000. The graph is in a U-shaped distribution, which means that the number of clients that get credit balance limit varies. It varies because as we see from graph a high number of client got 40,000 balance limit, low number of client got 150,000, high number of client got 200,000 balance limit and so on..

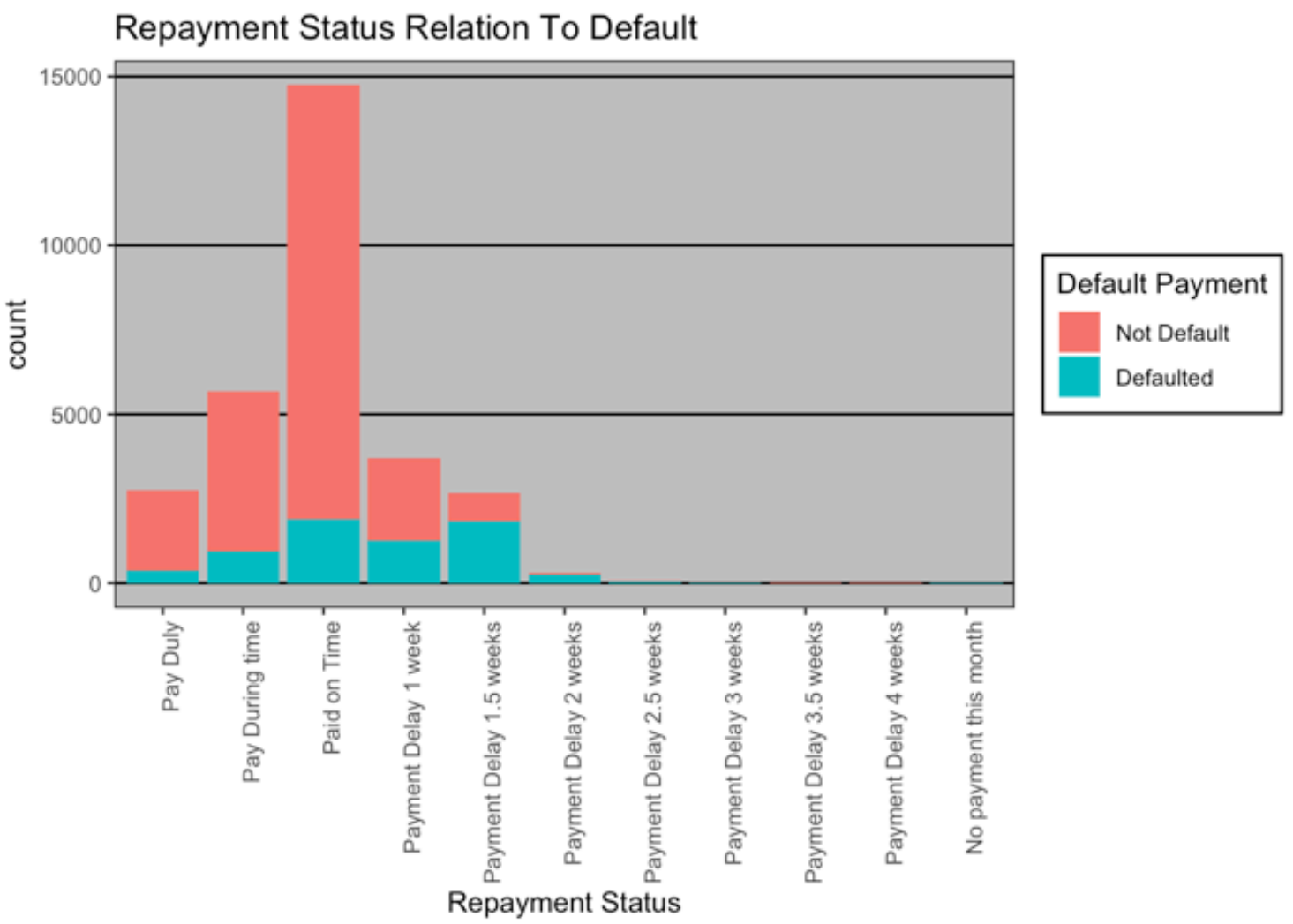
```
                                    <center>     Default Correlation
With The Factors </center>
```

```
ggplot(data, aes(x = EDUCATION, fill = default.payment.next.month))
+
  geom_bar() +
  labs(x = 'EDUCATION') +
  theme_economist_white()+
  ggtitle("Education Relation To Default")+ scale_x_discrete(labels=
ed)+scale_fill_discrete(labels=def)+ labs(fill='Default Payment')
```



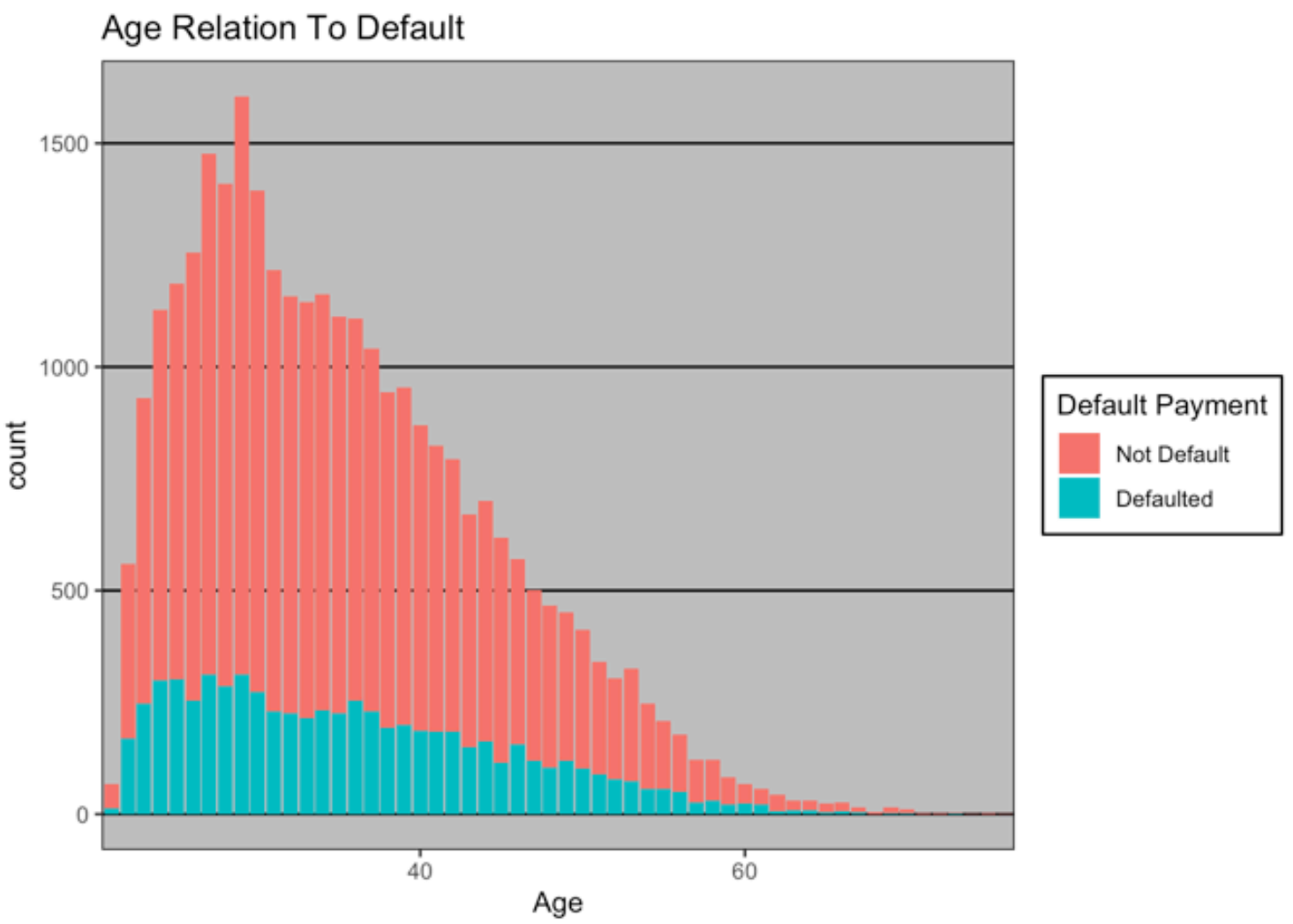
From this graph, We see that closer to 1/3 of the people that are in university have credit default. Also, about 1/4 of the grad student have credit default. As for the high school students, about 1/3 of them have credit default. Neither the people that have G.E.D or people that marked other as their education have credit default since there isn't so many of them. From this graph, we conclude that the higher education people have, the amount of default get to be less.

```
ggplot(data, aes(x = PAY_0, fill = default.payment.next.month)) +
  geom_bar() +
  labs(x = 'Repayment Status') +
  theme_excel()+
  ggtitle("Repayment Status Relation To Default")+scale_fill_discret
e(labels=def)+scale_x_discrete(labels=repa)+ theme(axis.text.x = ele
ment_text(angle = 90, hjust = 1))+ labs(fill='Default Payment')
```



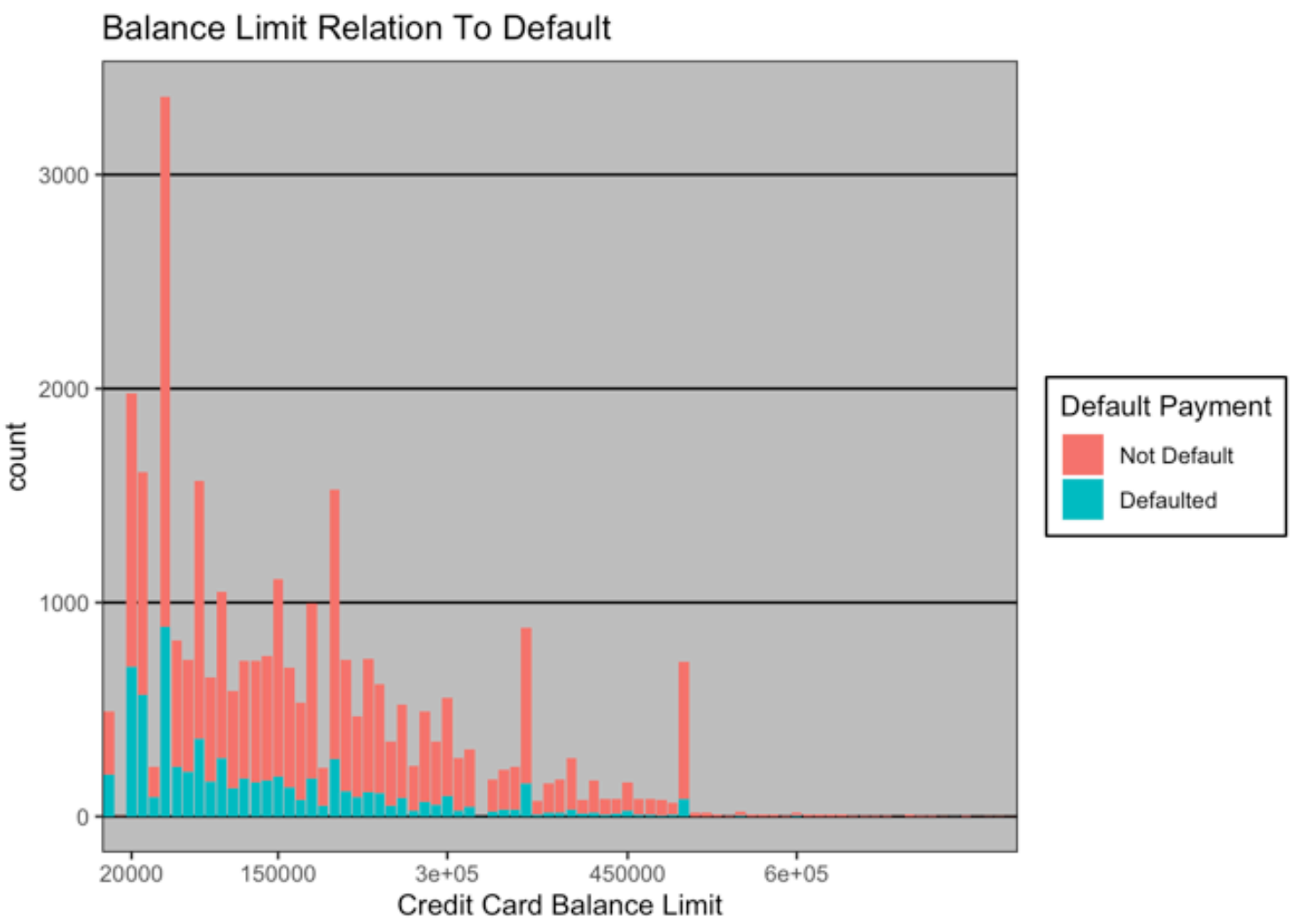
From this graph, we can see that as some many people made their payment on time. Also, from the graph, we can see that there is a huge number of people that got defaulted due the fact that they delayed their payment for two month. As the people delay their payments for 3month, the majority of them get default on their credit card. So from this graph, the more people delay their payment, the more people get default.

```
ggplot(A, aes(x = AGE, fill = default.payment.next.month)) +
  geom_bar() +
  labs(x = 'Age') +
  theme_excel()+
  ggtitle("Age Relation To Default")+scale_x_discrete( breaks=c(20,4
0,60,80),) +scale_fill_discrete(labels=def)+ labs(fill='Default Paym
ent')
```



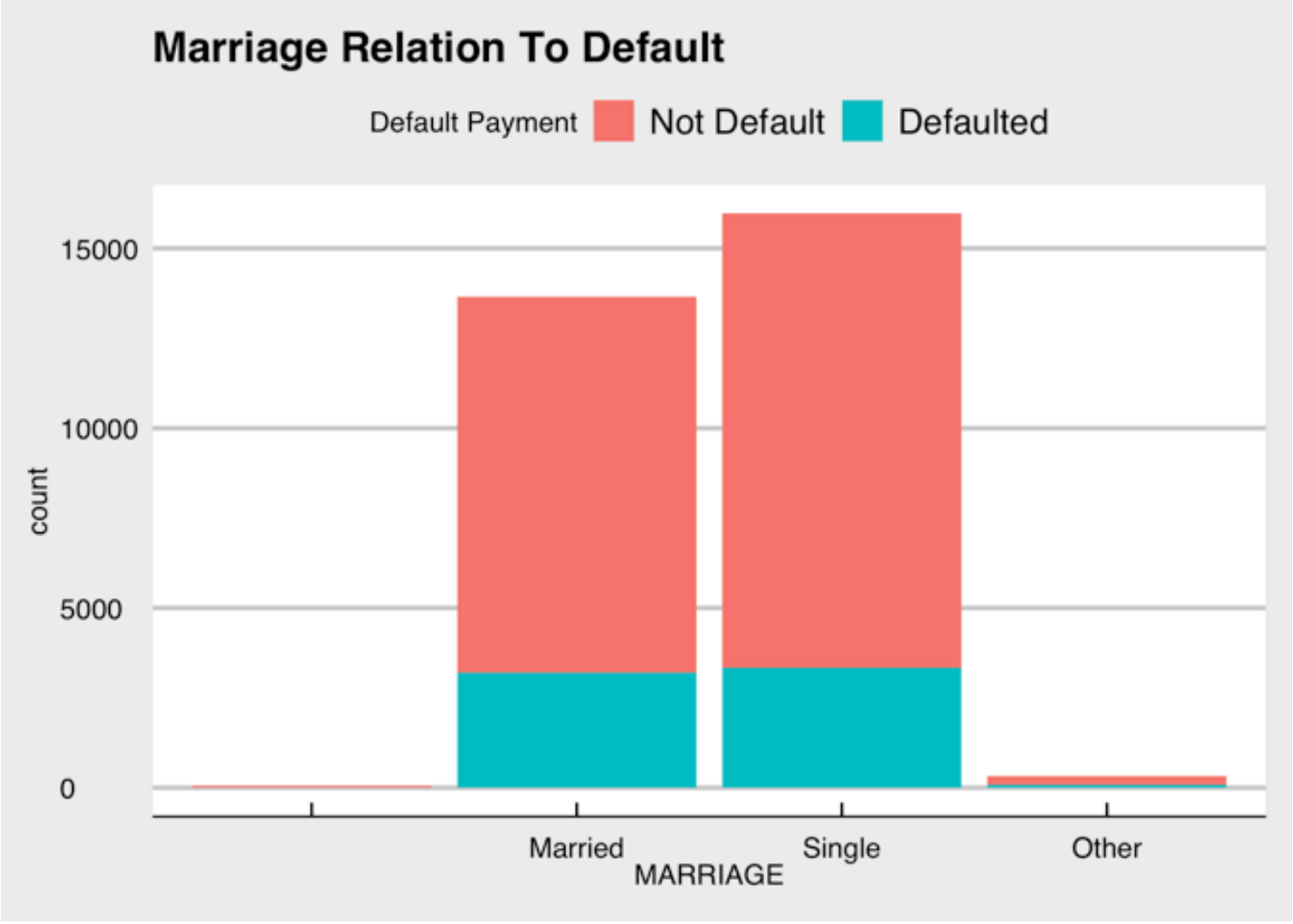
From this graph, we can see that a high number of people between 20 and 30 applid for a loan. As people reaches 50 years old or higher, the number of people that have default on their credit card get to increase. From this graph, we conclude that the number of default stays constant as the number of non-default decreases, and age increases

```
ggplot(data, aes(x = LIMIT_BAL, fill = default.payment.next.month))
+
  geom_bar() +
  labs(x = 'Credit Card Balance Limit') +
  theme_excel()+
  ggtitle("Balance Limit Relation To Default")+
  scale_x_discrete( breaks=c( 20000, 150000, 300000,450000,600000),)
+scale_fill_discrete(labels=def)+ labs(fill='Default Payment')
```



As we see from this graph, people that got 20,000 credit, closer to half of them got defaulted. As the credit limit increases, the number of people that get defaulted decreases. From the graph, as the balance limit increases, the number of client increases then decreases and so on..

```
ggplot(data, aes(x = MARRIAGE, fill = default.payment.next.month),)
+
  geom_bar() +
  labs(x = 'MARRIAGE') +
  theme_economist_white()+
  ggtitle("Marriage Relation To Default")+ scale_x_discrete(labels=
Mar)+scale_fill_discrete(labels=def)+ labs(fill='Default Payment')
```



From this graph, we can see that the amount of single and married people that got credit default are closely the same. The only diffrence we see is that there is huge single people compared to married people. As for the people that have other as their marital status, about of half of them recieved a credit default. From this graph, as people get married, the number of default decrease.

logistic regression

From those logistic regressions, we try to use logistic regression to explain the best

```
AgeReg = glm(formula=default.payment.next.month~AGE, data=A,family =
binomial)
predict.prob= predict(AgeReg,type="response")
pred.glm=rep(0, length(predict.prob))
pred.glm[predict.prob>0.5]=1
predict.table=table(pred.glm,A$default.payment.next.month)
predict.table
```

```
##
## pred.glm     0     1
##         0 23363  6633
##         1     1     3
```

```
paste("The accuracy rate is ",mean(pred.glm==A$default.payment.next.
month) )
```

```
## [1] "The accuracy rate is  0.778866666666667"
```

As this regression have nice accuracy rate 0.778, we will still search for a regression that has a better accuracy rate.

```
LimitedbalReg = glm(formula=default.payment.next.month~LIMIT_BAL, da
ta=A,family = binomial)
predict.prob2= predict(LimitedbalReg,A,type="response")
pred.glm2=rep(0, length(predict.prob2))
pred.glm2[predict.prob2>0.5]=1
predict.table2=table(pred.glm2,A$default.payment.next.month)
predict.table2
```

```
##
## pred.glm2     0     1
##          0 23363  6634
##          1     1     2
```

```
paste("The accuracy rate is ",mean(pred.glm2==A$default.payment.next
.month) )
```

```
## [1] "The accuracy rate is  0.778833333333333"
```

Again, another regression that is similar to the limited balance regresssion. It has high accuracy rate but not the highest.

```
BillamReg = glm(formula=default.payment.next.month~PAY_2, data=A,fam
ily = binomial)
predict.prob4= predict(BillamReg,A,type="response")
pred.glm4=rep(0, length(predict.prob4))
pred.glm4[predict.prob4>0.5]=1
predict.table4=table(pred.glm4,A$default.payment.next.month)
predict.table4
```

```
##
## pred.glm4     0     1
##         0 21426  4165
##         1  1938  2471
```

```
paste("The accuracy rate is ",mean(pred.glm4==A$default.payment.next
.month) )
```

```
## [1] "The accuracy rate is  0.796566666666667"
```

The accuracy rate for the payment status regression for the 2nd month is high, but there is a higher accuracy regression that we need to find.

```
PayamtReg = glm(formula=default.payment.next.month~PAY_3, data=A,fam
ily = binomial)
predict.prob3= predict(PayamtReg,A,type="response")
pred.glm3=rep(0, length(predict.prob3))
pred.glm3[predict.prob3>0.5]=1
predict.table3=table(pred.glm3,A$default.payment.next.month)
predict.table3
```

```
##
## pred.glm3     0     1
##         0 21356  4435
##         1  2008  2201
```

```
paste("The accuracy rate is ",mean(pred.glm3==A$default.payment.next
.month) )
```

```
## [1] "The accuracy rate is  0.785233333333333"
```

As we can see, the payment status regression for the 3nd month is lower than the payment status regression for the 2nd month, therefore, we won't pick this regression to explain.

```
PaymentReg = glm(formula=default.payment.next.month~PAY_0, data=A,fa
mily = binomial)
predict.prob1= predict(PaymentReg,A,type="response")
pred.glm1=rep(0, length(predict.prob1))
pred.glm1[predict.prob1>0.5]=1
predict.table1=table(pred.glm1,A$default.payment.next.month)
predict.table1
```

```
##
## pred.glm1     0     1
##         0 22411  4459
##         1   953  2177
```

```
paste("The accuracy rate is ",mean(pred.glm1==A$default.payment.next
.month) )
```

```
## [1] "The accuracy rate is  0.8196"
```

As we can see, this payment status logistics regression has the highest accuracy rate of all other customers characterisitcs. Therefore, we are interpret this regression below.

```
paste("The true Positive rate is", (22411)/(22411+4459) )
```

```
## [1] "The true Positive rate is 0.83405284704131"
```

```
paste("The false Positive rate is", (4459)/(22411+4459) )
```

```
## [1] "The false Positive rate is 0.16594715295869"
```

From this, we can say that 83% of this regression prediction is true and accurate. As only 17% of the regression prediction is unaccurate.

```
summary(PayamtReg)
```

```
##
## Call:
## glm(formula = default.payment.next.month ~ PAY_3, family = binomi
al,
##     data = A)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8365  -0.6193  -0.6193  -0.5823   1.9278
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.48076    0.04027 -36.773  < 2e-16 ***
## PAY_3-1     -0.20795    0.05386  -3.861 0.000113 ***
## PAY_30      -0.07322    0.04541  -1.613 0.106847
## PAY_31       0.38215    1.15540   0.331 0.740832
## PAY_32       1.54310    0.05167  29.864  < 2e-16 ***
## PAY_33       1.78304    0.13664  13.049  < 2e-16 ***
## PAY_34       1.79922    0.23579   7.630 2.34e-14 ***
## PAY_35       1.76845    0.44279   3.994 6.50e-05 ***
## PAY_36       1.92260    0.42914   4.480 7.46e-06 ***
## PAY_37       2.96237    0.49707   5.960 2.53e-09 ***
## PAY_38       2.17391    1.22541   1.774 0.076058 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 31705  on 29999  degrees of freedom
## Residual deviance: 29469  on 29989  degrees of freedom
## AIC: 29491
##
## Number of Fisher Scoring iterations: 4
```

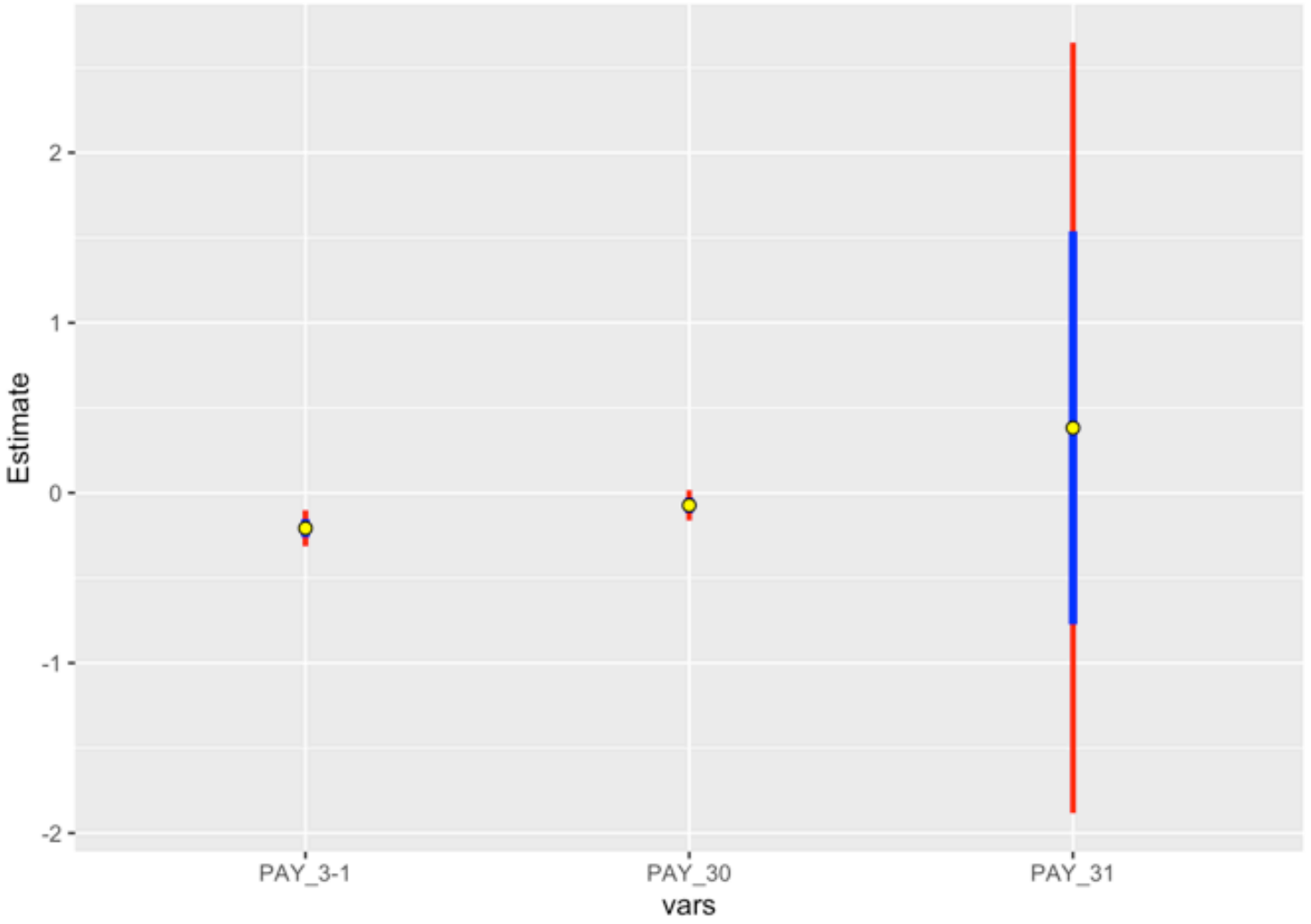
From this regression, you can see how payments have impacts on the number of defaults. Also, From this regression, we can see that as payments are made on time, default decreases by -1.48076. We can see that first 2 payments were delayed which caused them to be insignificant as the other payment are significant.

Significance Test

```
lm.summary = summary(PayamtReg)
lm.summary$coefficient
```

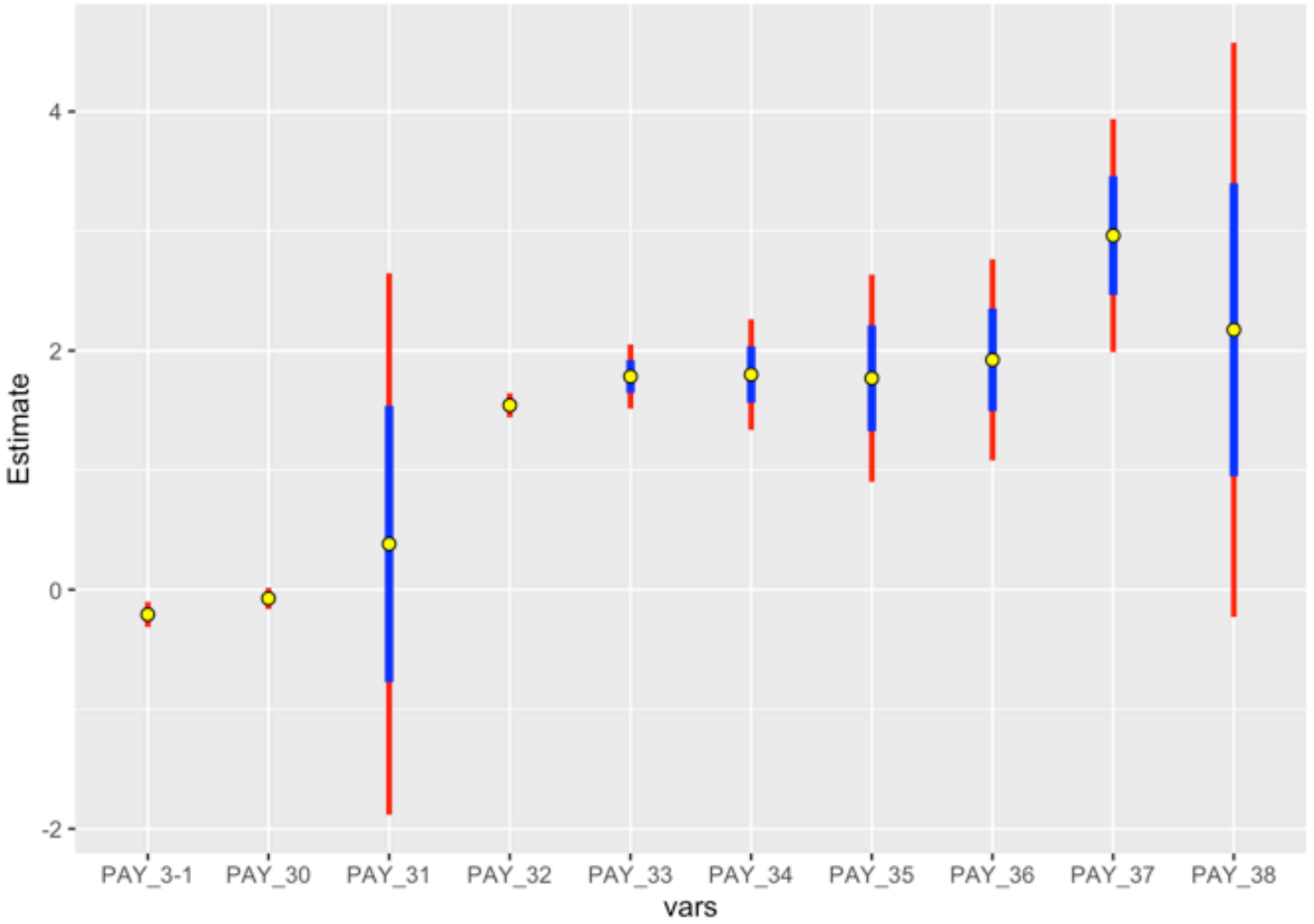
```
##                Estimate Std. Error     z value      Pr(>|z|)
## (Intercept) -1.48076355 0.04026766 -36.7730211 4.984602e-296
## PAY_3-1     -0.20795253 0.05386018  -3.8609697  1.129379e-04
## PAY_30      -0.07322083 0.04540743  -1.6125296  1.068468e-01
## PAY_31       0.38215126 1.15540145   0.3307519  7.408319e-01
## PAY_32       1.54310371 0.05167105  29.8639916 5.778350e-196
## PAY_33       1.78304442 0.13664473  13.0487607  6.458947e-39
## PAY_34       1.79921728 0.23579389   7.6304662  2.339063e-14
## PAY_35       1.76844562 0.44279333   3.9938398  6.501179e-05
## PAY_36       1.92259630 0.42914003   4.4801141  7.460314e-06
## PAY_37       2.96236809 0.49706740   5.9596910  2.527154e-09
## PAY_38       2.17391073 1.22540644   1.7740324  7.605779e-02
```

```
coefs = as.data.frame(lm.summary$coefficients[2:4,1:2])
names(coefs)[2] = "se"
coefs$vars = rownames(coefs)
ggplot(coefs, aes(vars, Estimate)) +
geom_errorbar(aes(ymin=Estimate - 1.96*se, ymax=Estimate + 1.96*se),
lwd=1, colour="red", width=0) +
geom_errorbar(aes(ymin=Estimate - se, ymax=Estimate + se), lwd=1.5,
colour="blue", width=0) +
geom_point(size=2, pch=21, fill="yellow")
```



As we can see from the graph, payment 31 is significant, while payment 29 and 30 were insignificant. It seems that payment 29 and 30 weren't paid on time as payment 31.

```
coefs = as.data.frame(lm.summary$coefficients[-1,1:2])
names(coefs)[2] = "se"
coefs$vars = rownames(coefs)
ggplot(coefs, aes(vars, Estimate)) +
geom_errorbar(aes(ymin=Estimate - 1.96*se, ymax=Estimate + 1.96*se),
lwd=1, colour="red", width=0) +
geom_errorbar(aes(ymin=Estimate - se, ymax=Estimate + se), lwd=1.5,
colour="blue", width=0) +
geom_point(size=2, pch=21, fill="yellow")
```



From this graph, we can tell that the first 30 payment were insignificant. We can tell that those payment are insignificant since these payment were below 0.05 significance level.

Probit Regression

```
PaymentProbit = glm(formula=default.payment.next.month~PAY_0, data=A
,family = binomial(link="probit"))
predict.prob5= predict(PaymentProbit,A,type="response")
pred.glm5=rep(0, length(predict.prob5))
pred.glm5[predict.prob5>0.5]=1
predict.table5=table(pred.glm5,A$default.payment.next.month)
predict.table5
```

```
##
## pred.glm5     0     1
##         0 22424  4472
##         1   940  2164
```

```
paste("The accuracy rate is ",mean(pred.glm5==A$default.payment.next
.month) )
```

```
## [1] "The accuracy rate is  0.8196"
```

As we done the probit regression, we can see that the probit regression's accuracy rate is still similar to the logistic regression's accuracy rate.

KNN

In the example below, we will use KNN method to find the percentage of customers that got defaulted.

```
Data_2 = A[c("default.payment.next.month")]
Data_1 = A[c("PAY_0", "AGE","LIMIT_BAL")]


test.X = Data_1 %>% slice(1:2000)
train.X = Data_1 %>% slice(2001:3000)
test.Y = Data_2 %>% slice(1:2000)
train.Y = Data_2 %>% slice(2001:3000)
train.Y=train.Y[,1, drop=TRUE]
set.seed(1)
knn_pred = knn(train.X, test.X, train.Y, k=1)

paste("Error rate is ", mean(knn_pred==A$default.payment.next.month)
)
```

```
## [1] "Error rate is  0.667166666666667"
```

```
paste("Percent got defaulted is ", mean(knn_pred != 0))
```

```
## [1] "Percent got defaulted is  0.2005"
```

Using the knn, we were able to figure out that only 20% of the people that applied for credit card got defaulted.