CV_Project_1

Group 11- Apeksha, Nithin, Prajwal

2022-02-28

DATA WRANGLING AND CLEANING

```
#Assigning all the column names in row 1 to it's main column
df_credit <- data_df %>%
  row_to_names(row_number = 1)

#Making the variables into categorical data as they are currently encoded

#Decoding the gender variables to categorical data
df_credit$SEX[df_credit$SEX == 1] <- "male"
df_credit$SEX[df_credit$SEX == 2] <- "female"

#Only 1,2 and 3 are the accepted values. There are 54 columns with 0 as the value
#Grouping these records into others

df_credit$MARRIAGE[df_credit$MARRIAGE == 1] <- "married"
df_credit$MARRIAGE[df_credit$MARRIAGE == 2] <- "single"
df_credit$MARRIAGE[df_credit$MARRIAGE == 3] <- "others"
df_credit$MARRIAGE[df_credit$MARRIAGE == 0] <- "others"


#Changing column names for our convenience
colnames(df_credit)[which(names(df_credit) == "PAY_0")] <- "repayment_status_september"
colnames(df_credit)[which(names(df_credit) == "PAY_2")] <- "repayment_status_august"
colnames(df_credit)[which(names(df_credit) == "PAY_3")] <- "repayment_status_july"
colnames(df_credit)[which(names(df_credit) == "PAY_4")] <- "repayment_status_june"
colnames(df_credit)[which(names(df_credit) == "PAY_5")] <- "repayment_status_may"
colnames(df_credit)[which(names(df_credit) == "PAY_6")] <- "repayment_status_april"


colnames(df_credit)[which(names(df_credit) == "BILL_AMT1")] <- "bill_september"
colnames(df_credit)[which(names(df_credit) == "BILL_AMT2")] <- "bill_august"
colnames(df_credit)[which(names(df_credit) == "BILL_AMT3")] <- "bill_july"
colnames(df_credit)[which(names(df_credit) == "BILL_AMT4")] <- "bill_june"
colnames(df_credit)[which(names(df_credit) == "BILL_AMT5")] <- "bill_may"
colnames(df_credit)[which(names(df_credit) == "BILL_AMT6")] <- "bill_april"


colnames(df_credit)[which(names(df_credit) == "PAY_AMT1")] <- "payment_september"
colnames(df_credit)[which(names(df_credit) == "PAY_AMT2")] <- "payment_august"
colnames(df_credit)[which(names(df_credit) == "PAY_AMT3")] <- "payment_july"
colnames(df_credit)[which(names(df_credit) == "PAY_AMT4")] <- "payment_june"
```

```
colnames(df_credit)[which(names(df_credit) == "PAY_AMT5")] <- "payment_may"
colnames(df_credit)[which(names(df_credit) == "PAY_AMT6")] <- "payment_april"

colnames(df_credit)[which(names(df_credit) == "default payment next month")] <- "df_pay"

colnames(df_credit)
```

```
##  [1] "ID"                         "LIMIT_BAL"
##  [3] "SEX"                        "EDUCATION"
##  [5] "MARRIAGE"                   "AGE"
##  [7] "repayment_status_september" "repayment_status_august"
##  [9] "repayment_status_july"      "repayment_status_june"
## [11] "repayment_status_may"       "repayment_status_april"
## [13] "bill_september"             "bill_august"
## [15] "bill_july"                  "bill_june"
## [17] "bill_may"                   "bill_april"
## [19] "payment_september"          "payment_august"
## [21] "payment_july"               "payment_june"
## [23] "payment_may"                "payment_april"
## [25] "df_pay"
```

```
#Checking for NA/ Missing values in the data
sum(is.na(df_credit))
```

```
## [1] 0
```

Findings: There are no missing values in the data

```
#Converting the data columns to numeric
df_credit$repayment_status_september <- as.numeric(df_credit$repayment_status_september)
df_credit$repayment_status_august <- as.numeric(df_credit$repayment_status_august)
df_credit$repayment_status_july <- as.numeric(df_credit$repayment_status_july)
df_credit$repayment_status_june <- as.numeric(df_credit$repayment_status_june)
df_credit$repayment_status_may <- as.numeric(df_credit$repayment_status_may)
df_credit$repayment_status_april <- as.numeric(df_credit$repayment_status_april)
```

```
#Checking for anomalous data

unique(df_credit$repayment_status_september)
```

```
##  [1]  2 -1  0 -2  1  3  4  8  7  5  6
```

```
#The only acceptable values for repayment status are -1 which indicates paid duly
#and the positive values, indicating the number of months the payment was delayed
#Example - The value 1 indicates that the payment was made 1 month late,
#2 indicates 2 month delay and so on

df_credit$repayment_status_september[df_credit$repayment_status_september == -2] <- "-1"

#According to the data if the positive values are the months of delayed payment,
#all the negative values should be grouped under 0 as duly paid
```

```r
#Modifying the data accordingly to group negative values under 0 which indicates duly paid
df_credit_1 <- df_credit %>% mutate(repayment_status_september =
                                    replace(repayment_status_september,
                                    repayment_status_september == -1 |
                                    repayment_status_september == -2  , 0))

df_credit_1 <- df_credit_1 %>% mutate(repayment_status_august =
                                      replace(repayment_status_august,
                                      repayment_status_august == -1 |
                                      repayment_status_august == -2  , 0))

df_credit_1 <- df_credit_1 %>% mutate(repayment_status_july =
                                      replace(repayment_status_july,
                                      repayment_status_july == -1 |
                                      repayment_status_july == -2  , 0))

df_credit_1 <- df_credit_1 %>% mutate(repayment_status_june =
                                      replace(repayment_status_june,
                                      repayment_status_june == -1 |
                                      repayment_status_june == -2  , 0))

df_credit_1 <- df_credit_1 %>% mutate(repayment_status_may =
                                      replace(repayment_status_may,
                                      repayment_status_may == -1 |
                                      repayment_status_may == -2  , 0))

df_credit_1 <- df_credit_1 %>% mutate(repayment_status_april =
                                      replace(repayment_status_april,
                                      repayment_status_april == -1 |
                                      repayment_status_april == -2  , 0))
```

```r
unique(df_credit_1$EDUCATION)
```

```
## [1] "2" "1" "3" "5" "4" "6" "0"
```

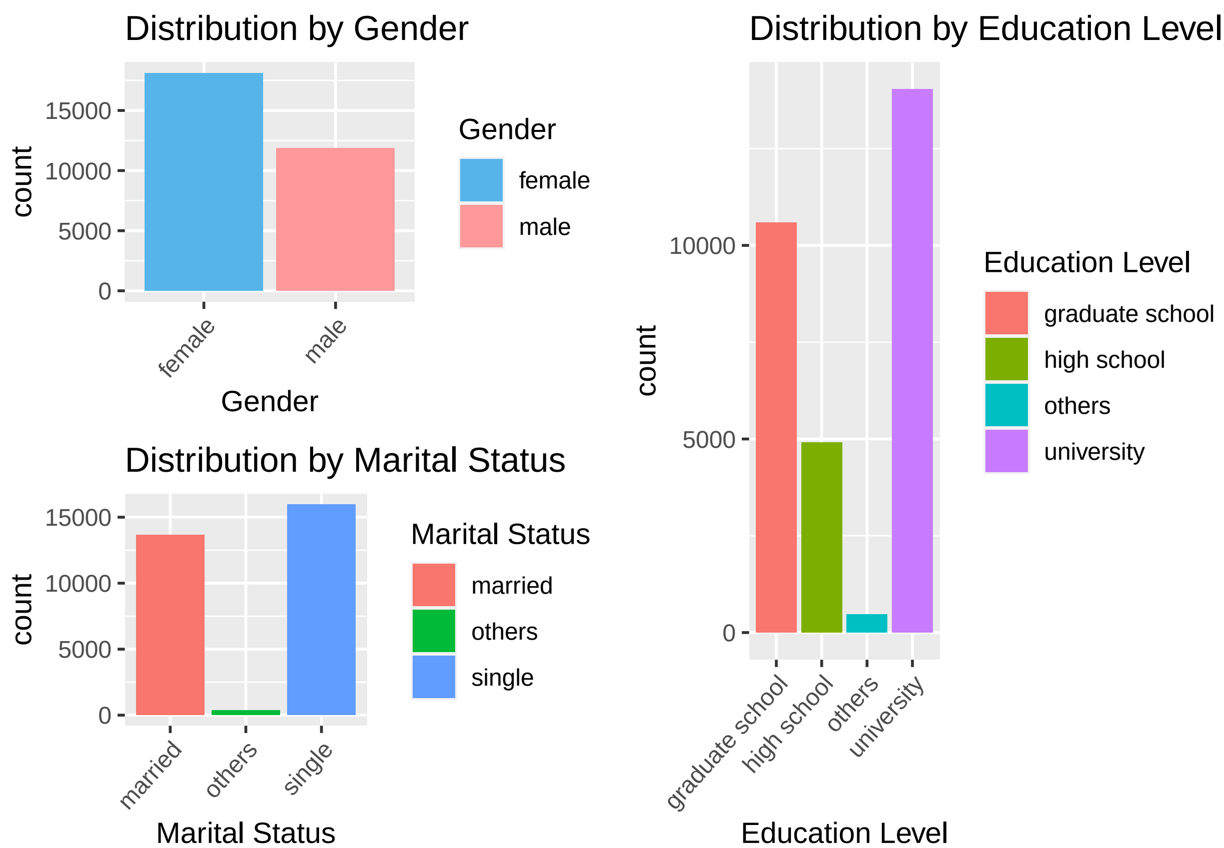
```r
# The only accepted values are 1, 2, 3 and 4. We have a few other values like 0,5 and 6
# Modifying these anomalous values into the others category to avoid loss of data

df_credit_1 <- df_credit %>% mutate(EDUCATION = replace(EDUCATION,
                                    EDUCATION == 0 | EDUCATION == 5 | EDUCATION == 6 , 4))

#All the anomalous values are being replaced by 4 as it indicates the others category

#Decoding the education level to different education levels of data

df_credit_1$EDUCATION[df_credit_1$EDUCATION == 1] <- "graduate school"
df_credit_1$EDUCATION[df_credit_1$EDUCATION == 2] <- "university"
df_credit_1$EDUCATION[df_credit_1$EDUCATION == 3] <- "high school"
df_credit_1$EDUCATION[df_credit_1$EDUCATION == 4] <- "others"

df_unique_edu <- unique(df_credit_1$EDUCATION)
```

All the data has been cleaned and brought into the right form with categorical columns and acceptable values

DATA VISUALIZATION

1. Total counts of the values that are present in each particular field

```
#Total Count with respect to Sex
graph1 <- ggplot(data=df_credit_1, aes(x=SEX,fill=SEX)) +
          geom_bar() +
          labs(title = "Distribution by Gender", x ="Gender",fill = "Gender") +
          scale_fill_manual(values=c("#56B4E9", "#FF9999")) +
          theme(axis.text.x = element_text(angle = 48,hjust=1))

#Total Count with respect to Education
graph2 <- ggplot(data=df_credit_1, aes(x=EDUCATION,fill=EDUCATION)) +
          geom_bar() +
          labs(title = "Distribution by Education Level",
               x ="Education Level",fill = "Education Level") +
          theme(axis.text.x = element_text(angle = 48,hjust=1))

#Total Count with respect to Marriage
graph3 <- ggplot(data=df_credit_1, aes(x=MARRIAGE,fill=MARRIAGE)) +
          geom_bar() +
          labs(title = "Distribution by Marital Status",
               x ="Marital Status",fill = "Marital Status") +
          theme(axis.text.x = element_text(angle = 48,hjust=1))

# Plotting data using combination of demographic attributes

grid.arrange( arrangeGrob(graph1,graph3, ncol=1),
arrangeGrob(graph2),
ncol=2, widths=c(1,1)) +
theme(axis.line = element_line(color='black'),
plot.background = element_blank(),
panel.grid.major = element_blank(),
panel.grid.minor = element_blank(),
panel.border = element_blank())
```

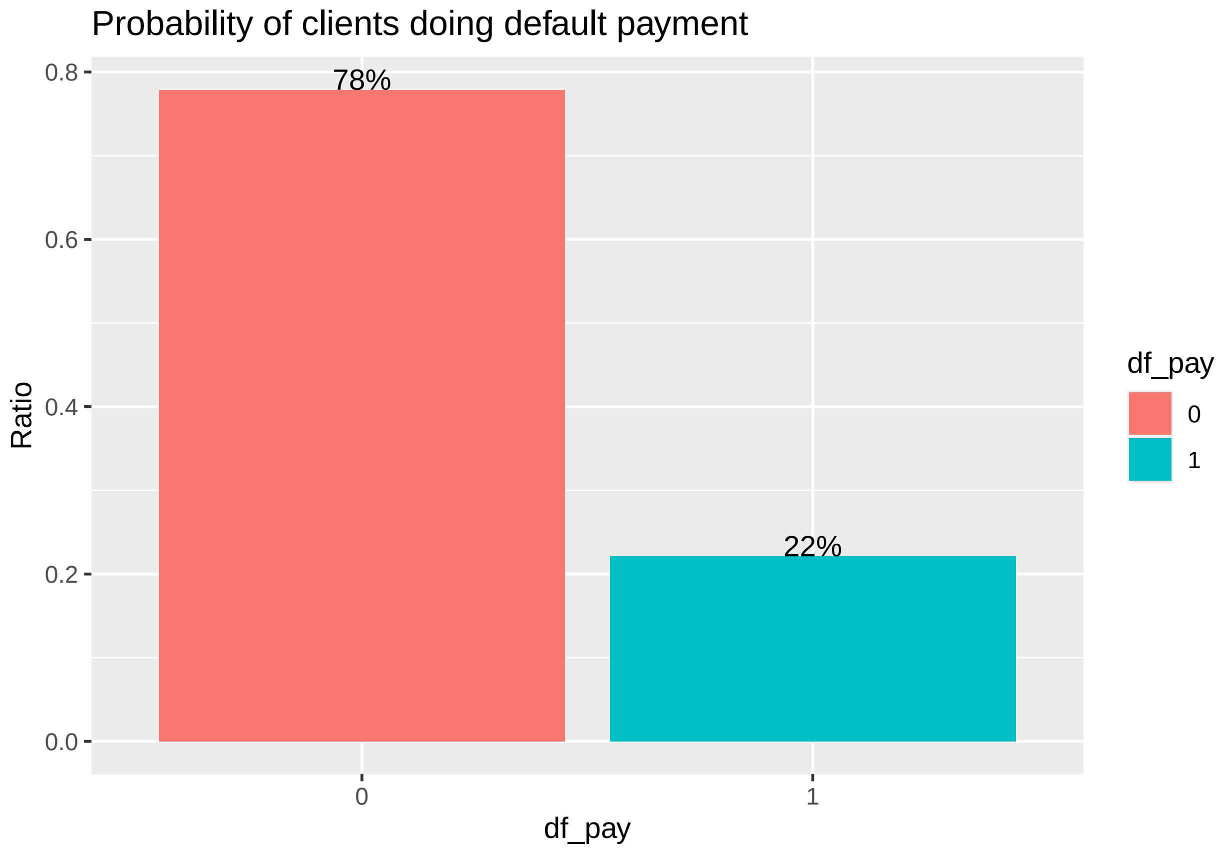



```
## NULL
```

Findings: It's observed that there are more female clients, with education level up to university and most of the clients are single followed by married

2. Probability of clients doing default payment in next month

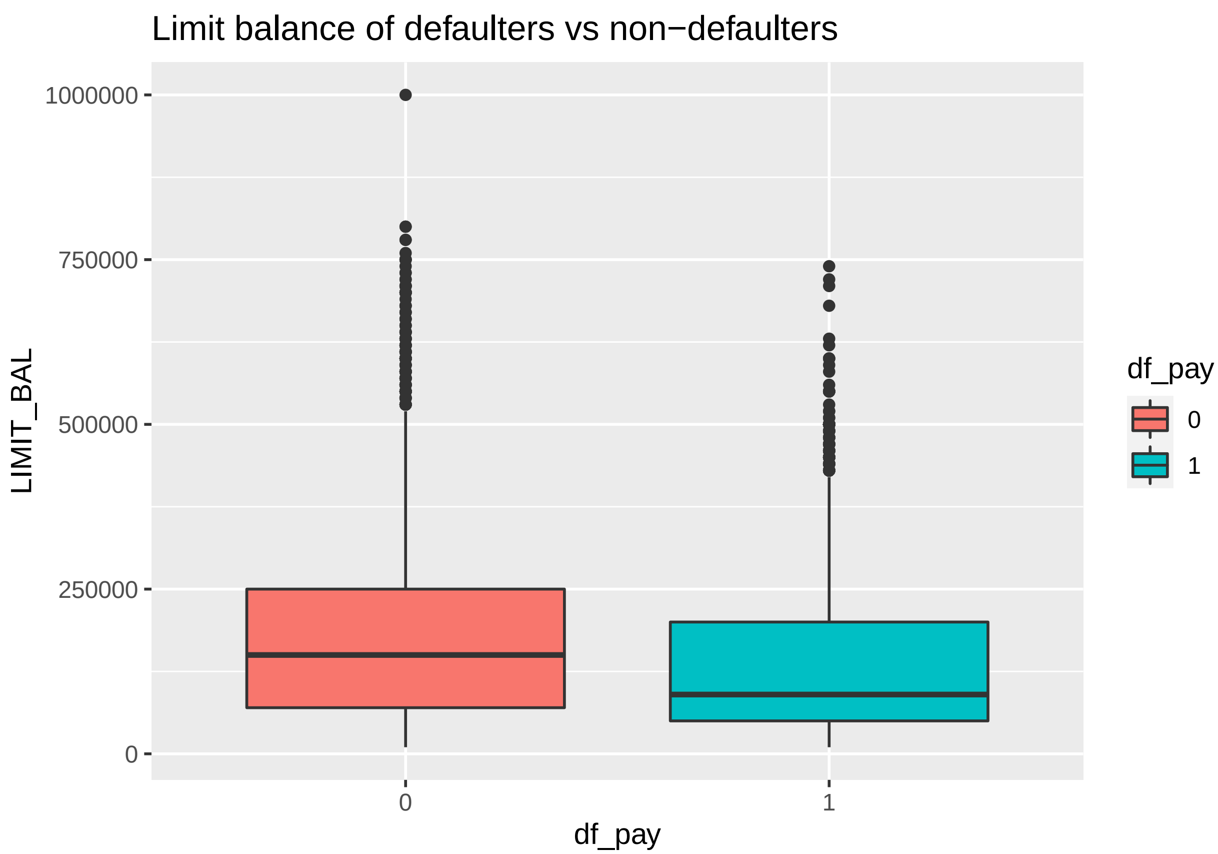
```
df_cent <- df_credit_1 %>%
  group_by(df_pay) %>%
  summarise(Count = n()) %>%
  mutate( df_pay = factor(df_pay),
    Ratio = Count / sum(Count),
         label = percent(Ratio %>% round(2)))

ggplot(df_cent, aes(x=df_pay,y=Ratio,label=label,fill=df_pay)) +
  geom_bar(stat='identity') +
  geom_text(vjust=0) +
  ggtitle("Probability of clients doing default payment")
```



Findings: We can see that the dataset consists of 78% clients that are not expected to default payment whereas 22% clients are expected to default the payment

3. Limit balance of defaulters vs non-defaulters

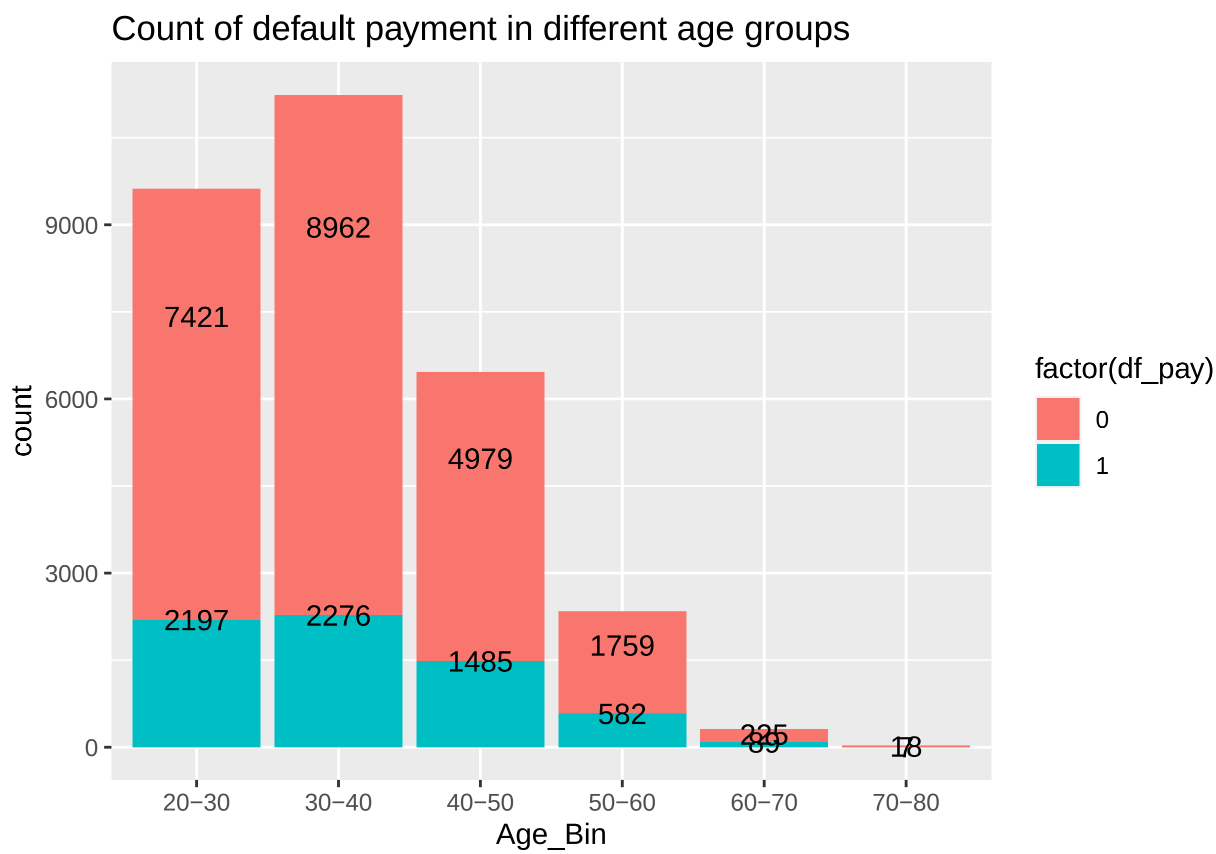
```
df_credit_1$LIMIT_BAL <- as.numeric(unlist(df_credit_1$LIMIT_BAL))

ggplot(data=df_credit_1,
       mapping = aes(x=df_pay,y=LIMIT_BAL,fill=df_pay)) +
       geom_boxplot()+
       ggtitle("Limit balance of defaulters vs non-defaulters")
```



Findings: The number of non-defaulters has more limit balance than defaulters

4. Age group as a factor to predict default payment for next month

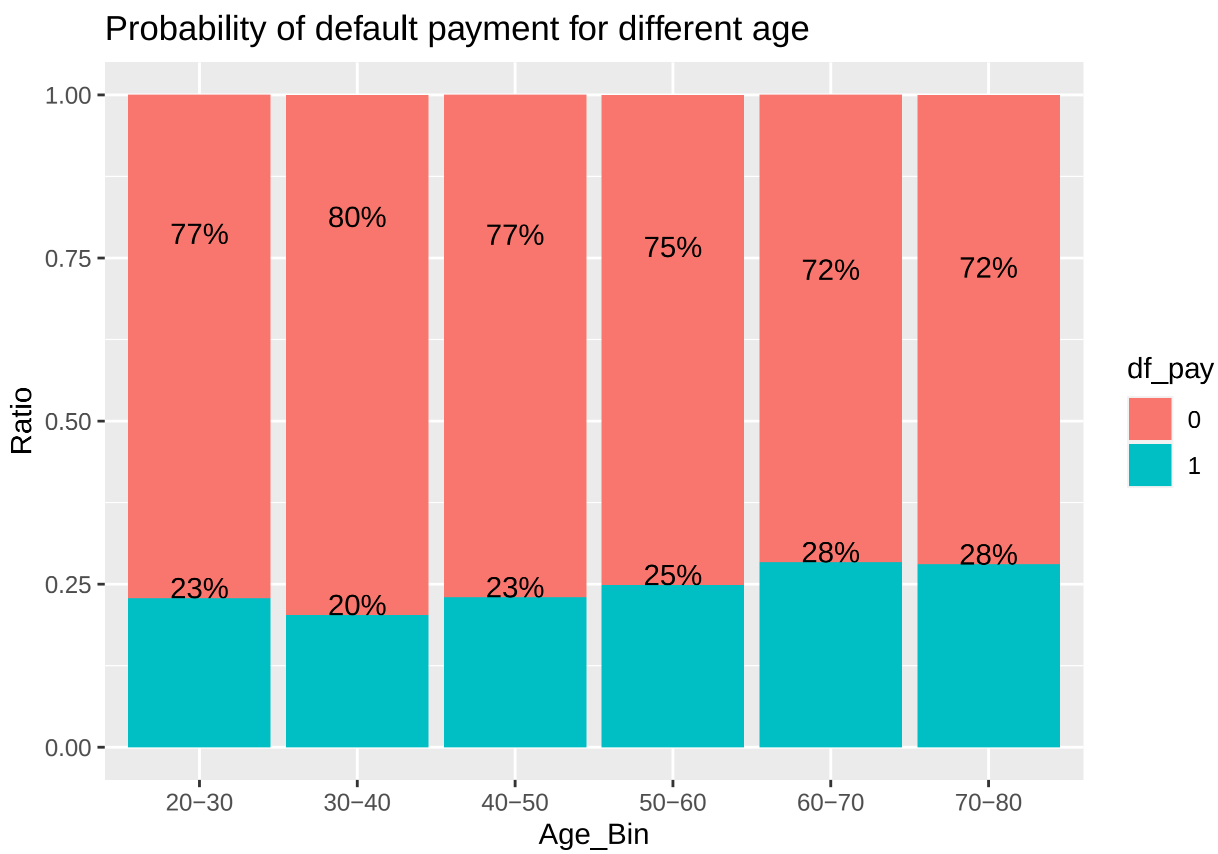
```
#Converting age column as numeric
df_credit_1$AGE <- as.numeric(df_credit$AGE)

#categorizing and labelling age into different bins
df_credit_1['Age_Bin'] <- cut(df_credit_1$AGE,
                              breaks=c(20, 30, 40, 50, 60, 70, 80),
                              right = FALSE,
                              labels = c("20-30", "30-40", "40-50", "50-60", "60-70", "70-80"))


ggplot(df_credit_1, aes(Age_Bin, fill = factor(df_pay))) +
  geom_bar(stat='count')+
  geom_text(aes(label = ..count..), stat = "count")+
  ggtitle("Count of default payment in different age groups")
```




```
#Finding the probability of defaulters across different ages
df_age_bin <- df_credit_1 %>%
  group_by(Age_Bin,df_pay) %>%
  summarise(Count = n()) %>%
  mutate( Age_Bin = factor(Age_Bin),
    Ratio = Count / sum(Count),
         label = percent(Ratio %>% round(2)))
```

```
## `summarise()` has grouped output by 'Age_Bin'. You can override using the `.groups` argument.
```

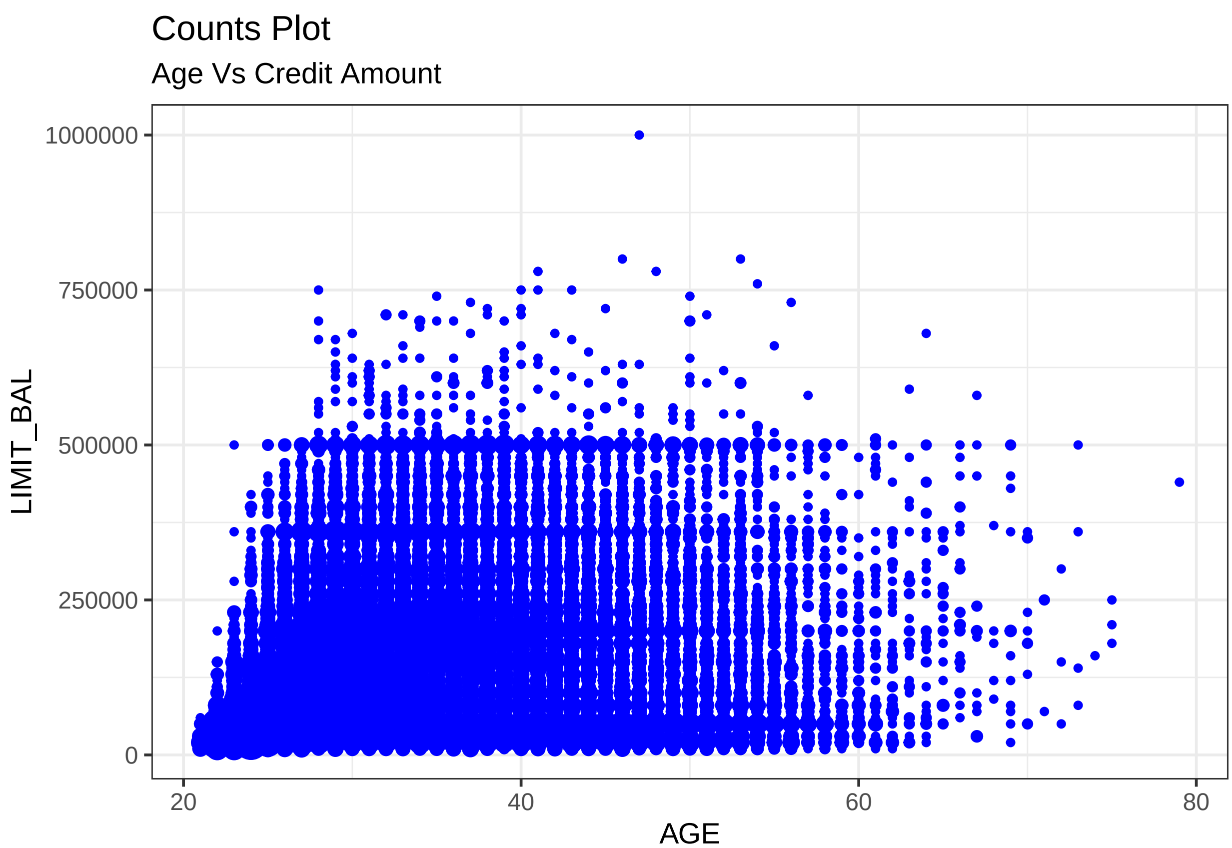
```
ggplot(df_age_bin, aes(x=Age_Bin,y=Ratio,label=label,fill=df_pay)) +
  geom_bar(stat='identity') +
  geom_text(vjust=0) +
  ggtitle("Probability of default payment for different age")
```



Findings: We have maximum clients from 21-30 age group followed by 31-40. Even though it looks like younger population has more defaulters, when we deep dived into the data we could observe that the older age groups have a higher percentage of defaulters than the younger ones

5. Age group with more Limit Balance

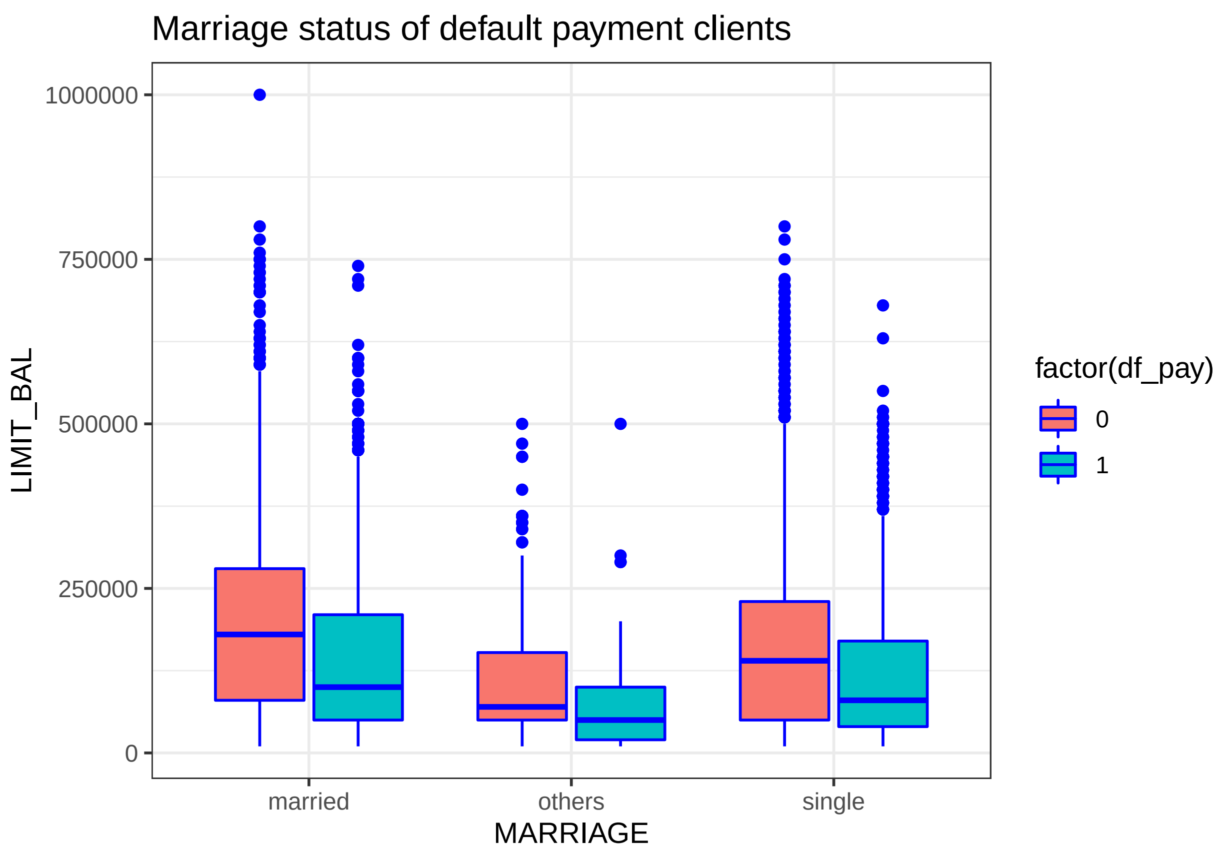
```
theme_set(theme_bw())
g <- ggplot(data = df_credit_1, aes(x = AGE, y = LIMIT_BAL))
g + geom_count(col="blue", show.legend=F) +
  labs(title="Counts Plot",
       subtitle="Age Vs Credit Amount", )
```



Findings: The credit limit is lesser for the youngest population and the oldest population when compared to the middle age groups

6. Marriage as a factor to predict default payment for next month

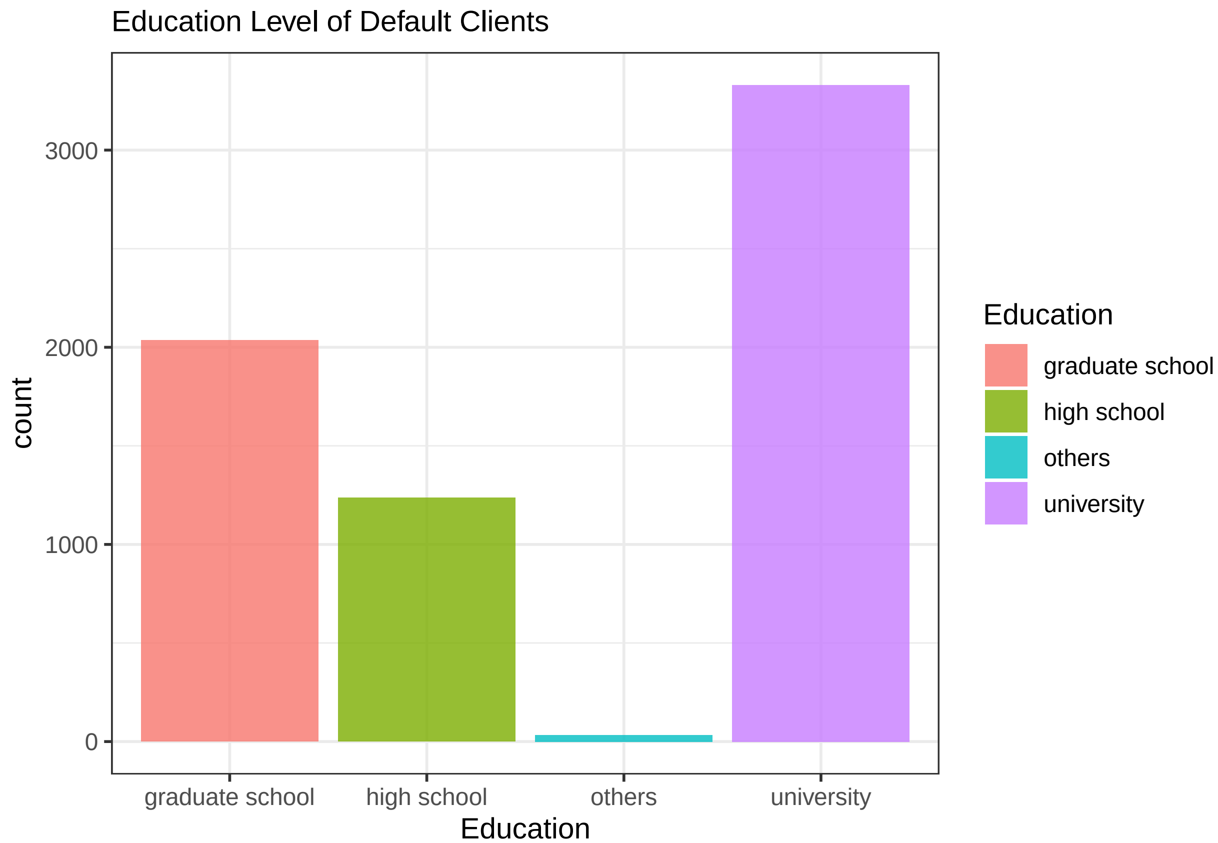
```
#Assigning df_credit_1 to new variable
df_credit_2 = df_credit_1

#converting default payment next month and education as a factor
df_credit_2$df_pay <- as.factor(df_credit_2$df_pay)

#Converting the new data frame columns to numeric columns
df_credit_2$LIMIT_BAL <- as.numeric(unlist(df_credit_2$LIMIT_BAL))

ggplot(data=df_credit_2,
       aes(x=MARRIAGE,y=LIMIT_BAL, fill = factor(df_pay))) +
       geom_boxplot(col = "blue")+
  ggtitle("Marriage status of default payment clients")
```



Findings:From the plot it can be observed that Non default payment clients who are married have more credit limit balance amount

7. Education Level of default payment clients

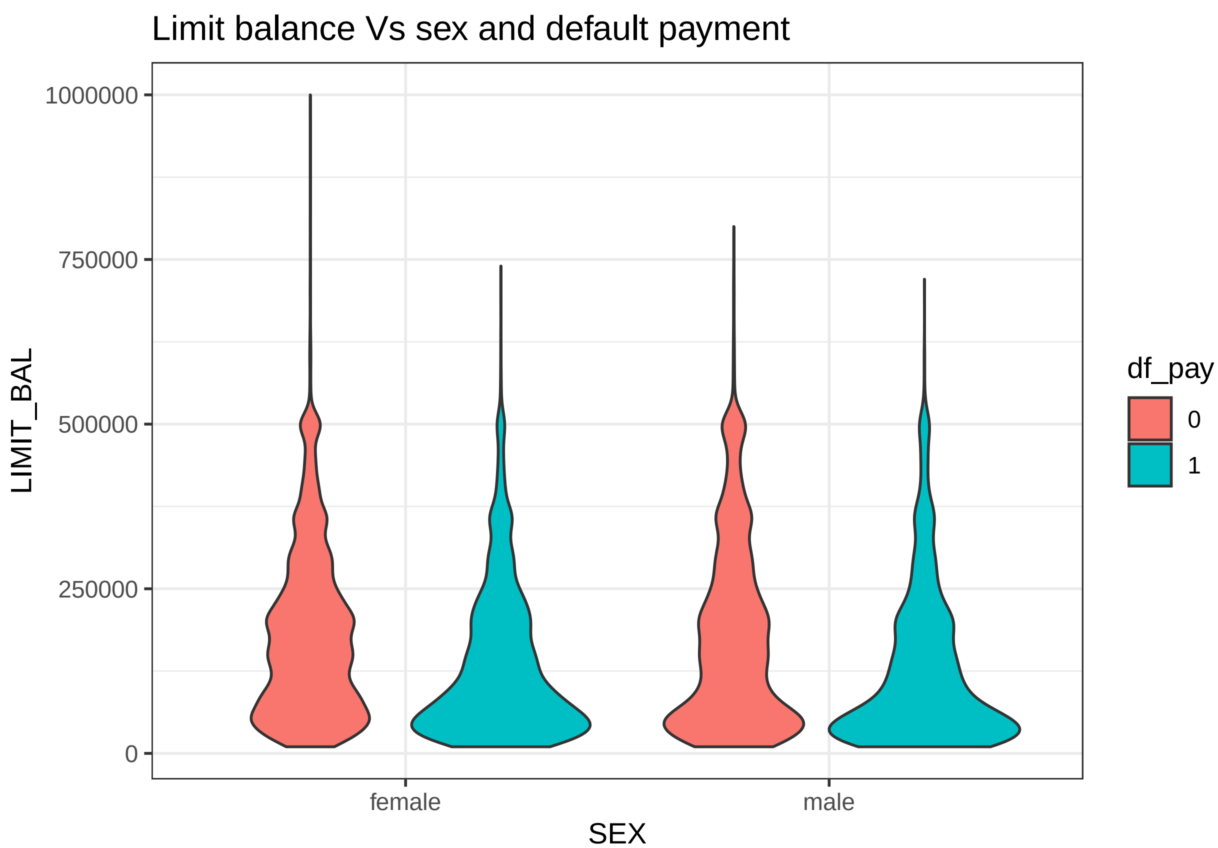
```
#Filtering default payment clients or clients that are expected to pay
df_credit_default <- filter(df_credit_1, df_credit_1$`df_pay` == '1')

ggplot(df_credit_default, aes(x=EDUCATION, fill = EDUCATION)) +
  geom_bar(aes(fill=factor(EDUCATION)), alpha=0.8) +
  labs(subtitle="Education Level of Default Clients",
       x="Education",
       fill="Education")
```



Findings: Most of the default payment clients are from University

8. Limit balance of both sex default and non default payees

```
ggplot(data = df_credit_1,
       aes(x = SEX, y = LIMIT_BAL, fill = df_pay))  +
       geom_violin()+
       ggtitle("Limit balance Vs sex and default payment")
```



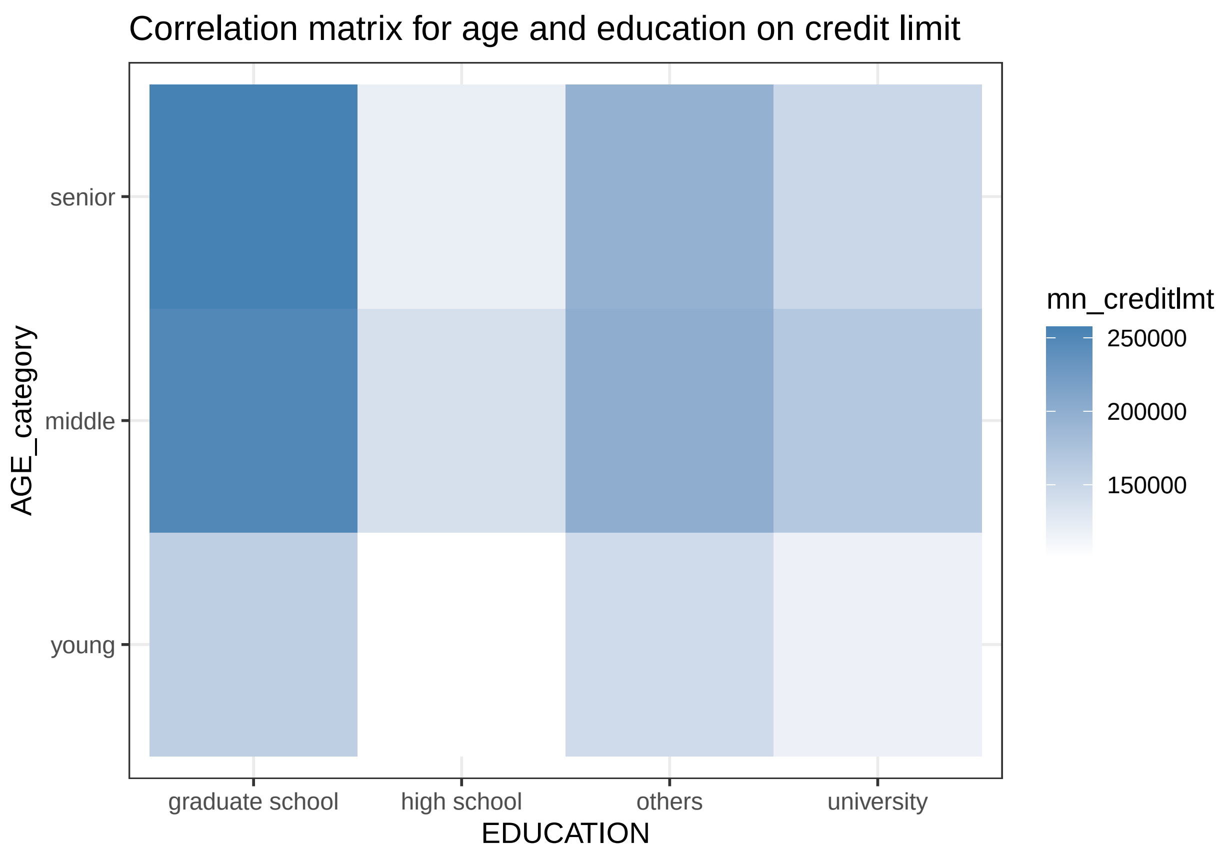
Findings: Most of the default payment clients are female with highest limit balance

9. Correlation between age and education based on credit limit

```
df_credit_1$AGE_category <- as.numeric(df_credit_1$AGE)
df_credit_1$AGE_category <-cut(df_credit_1$AGE_category,
                                breaks = c( 10, 30,50,100),
                                labels = c("young", "middle","senior"))

df_corr <- df_credit_1 %>% group_by(EDUCATION,AGE_category) %>%
  summarise(mn_creditlmt=mean(LIMIT_BAL))
```

```
## `summarise()` has grouped output by 'EDUCATION'. You can override using the `.groups` argument.
```

```
ggplot(df_corr, aes(EDUCATION, AGE_category, fill=mn_creditlmt)) +
  geom_tile() + scale_fill_gradient(low="white", high="steelblue")+
  ggtitle("Correlation matrix for age and education on credit limit")
```



Findings: Customers who completed graduate school and were in the middle and senior age groups seem to have a higher credit balance. This could be because customers with graduate degrees are more likely to have better paying jobs. This in turn helps them build better reputation with banks leading to higher credit limit

10. Credit card bill and actual payment for 6 months

```
#Converting the data columns to numeric

df_credit_1$bill_september <- as.numeric(df_credit$bill_september)
df_credit_1$bill_august <- as.numeric(df_credit$bill_august)
df_credit_1$bill_july <- as.numeric(df_credit$bill_july)
df_credit_1$bill_june <- as.numeric(df_credit$bill_june)
df_credit_1$bill_may <- as.numeric(df_credit$bill_may)
df_credit_1$bill_april <- as.numeric(df_credit$bill_april)


df_credit_1$payment_september <- as.numeric(df_credit$payment_september)
df_credit_1$payment_august <- as.numeric(df_credit$payment_august)
df_credit_1$payment_july <- as.numeric(df_credit$payment_july)
df_credit_1$payment_june <- as.numeric(df_credit$payment_june)
df_credit_1$payment_may <- as.numeric(df_credit$payment_may)
df_credit_1$payment_april <- as.numeric(df_credit$payment_april)

# Calculate the mean of bill sent to customers each month
monthly_bill <- df_credit_1 %>%
  summarise(bill_1 = mean(bill_september),
```

```r
            bill_2 = mean(bill_august),
            bill_3 = mean(bill_july),
            bill_4 = mean(bill_june),
            bill_5 = mean(bill_may),
            bill_6 = mean(bill_april))

monthly_bill <- as.data.frame((t(monthly_bill)))

# rename column
names(monthly_bill)[1] <- "avg.bill"

# rename row
row.names(monthly_bill) <- c("month_1","month_2","month_3","month_4","month_5","month_6")

monthly_bill
```

```
##         avg.bill
## month_1 51223.33
## month_2 49179.08
## month_3 47013.15
## month_4 43262.95
## month_5 40311.40
## month_6 38871.76
```

```r
# Calculate the mean of payment received from customers each month
monthly_pay <- df_credit_1 %>%
  summarise(month_1 = mean(payment_september),
            month_2 = mean(payment_august),
            month_3 = mean(payment_july),
            month_4 = mean(payment_june),
            month_5 = mean(payment_may),
            month_6 = mean(payment_april))

monthly_pay <- as.data.frame((t(monthly_pay)))

# rename column
names(monthly_pay)[1] <- "avg.payment"

monthly_pay
```

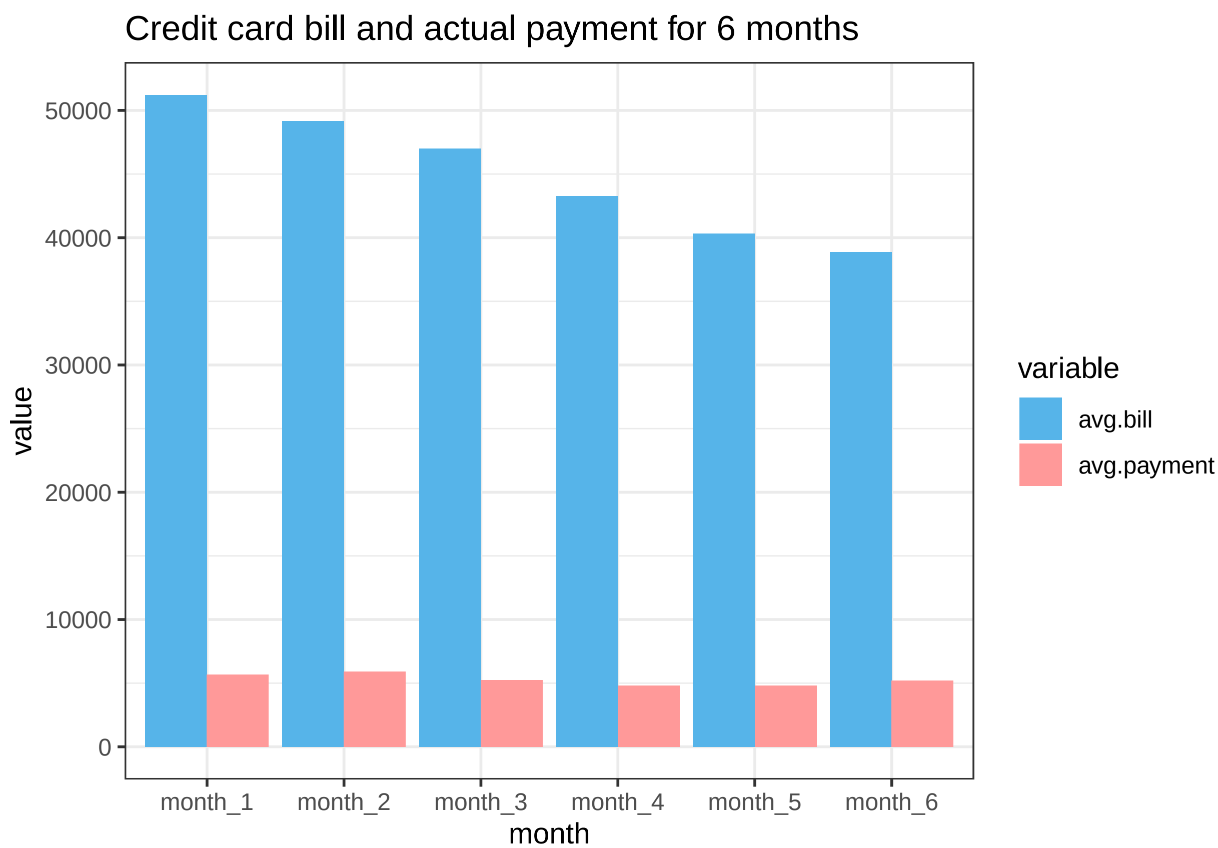
```
##         avg.payment
## month_1    5663.581
## month_2    5921.163
## month_3    5225.681
## month_4    4826.077
## month_5    4799.388
## month_6    5215.503
```

```r
# combine both bill & payment data
compare <- cbind(monthly_bill, monthly_pay)

# convert rownames into colnames
```

```r
library(data.table)
setDT(compare, keep.rownames = "month")
compare

# use `melt` to make change table formats
compare2 <- melt(compare, id.vars = "month")
compare2
```

```
##       month    variable     value
##  1: month_1    avg.bill 51223.331
##  2: month_2    avg.bill 49179.075
##  3: month_3    avg.bill 47013.155
##  4: month_4    avg.bill 43262.949
##  5: month_5    avg.bill 40311.401
##  6: month_6    avg.bill 38871.760
##  7: month_1 avg.payment  5663.581
##  8: month_2 avg.payment  5921.163
##  9: month_3 avg.payment  5225.681
## 10: month_4 avg.payment  4826.077
## 11: month_5 avg.payment  4799.388
## 12: month_6 avg.payment  5215.503
```

```r
# create comparison graph
ggplot(compare2, aes(x=month, y=value, fill=variable))+
  geom_bar(stat = "identity", position = "dodge")+
    scale_fill_manual(values=c("#56B4E9", "#FF9999"))+
    ggtitle("Credit card bill and actual payment for 6 months")
```

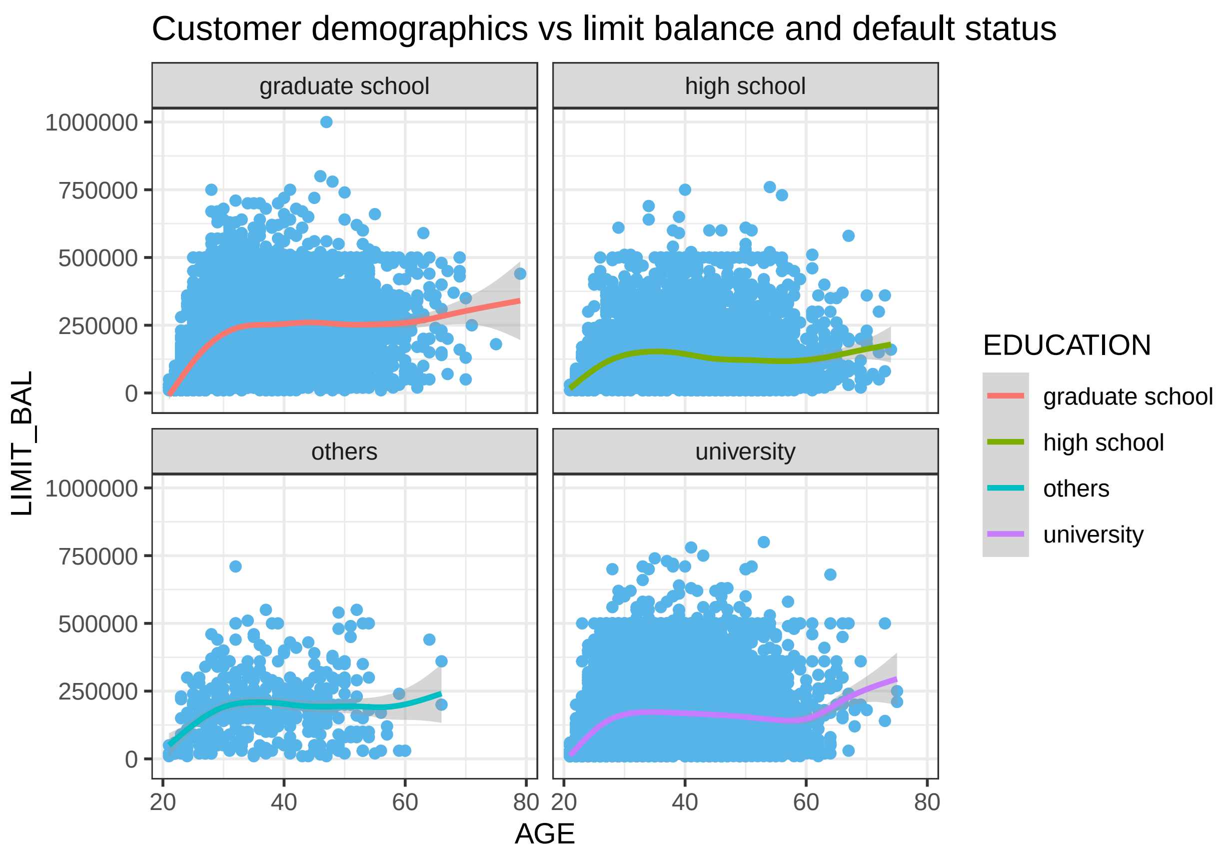


Findings: The bill is steadily decreased on following months, which makes sense because it was being paid within each month by the customer. The payment proportion is very small compared to the bill, which may be caused by the customer that chose longer installment term so the amount that they have to pay each month is small

11. Exploration of customer profile to look at how demographic factors reflect on their limit balance and default status

```
df_credit_1$LIMIT_BAL <- as.numeric(df_credit$LIMIT_BAL)

ggplot(data = df_credit_1, mapping = aes(x=AGE, y=LIMIT_BAL))+
  geom_point(col = "#56B4E9")+
  geom_smooth(aes(color = EDUCATION))+
  ggtitle("Customer demographics vs limit balance and default status")+
  facet_wrap(~EDUCATION)
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



Findings : we can see that customer from "Graduate School (GS)" and "University (Uni)" have similar limit balance pattern (with the former has slightly higher population). Our main customer from GS & Uni background has similar age bracket too, around mid-20s to 50s. And since our data doesn't provide what "Others" Education is, we can ignore it and just focus on the 3 (GS, Uni, HS)

12. Variation of bills and payments for customers and defualters in 6 months

```
# Calculate the mean of bill sent to customers each month
monthly_bill_cust <- df_credit_1 %>%  group_by(df_pay) %>%
  summarise(bill_1 = mean(bill_september),
            bill_2 = mean(bill_august),
            bill_3 = mean(bill_july),
            bill_4 = mean(bill_june),
            bill_5 = mean(bill_may),
            bill_6 = mean(bill_april))

monthly_bill_cust <- as.data.frame((t(monthly_bill_cust)))

# rename column
names(monthly_bill_cust)[1] <- "avg.bill"

names(monthly_bill_cust) <- monthly_bill_cust[1,]
monthly_bill_cust <- monthly_bill_cust[-1,]

# rename row
row.names(monthly_bill_cust) <- c("month_1","month_2","month_3","month_4","month_5","month_6")
```

```
monthly_bill_cust <-monthly_bill_cust %>%
                      rename(
                        customer_bill = '0'  ,
                        defaulter_bill = '1'
                        )

monthly_bill_cust
```

```
##         customer_bill defaulter_bill
## month_1      51994.23       48509.16
## month_2      49717.44       47283.62
## month_3      47533.37       45181.60
## month_4      43611.17       42036.95
## month_5      40530.45       39540.19
## month_6      39042.27       38271.44
```

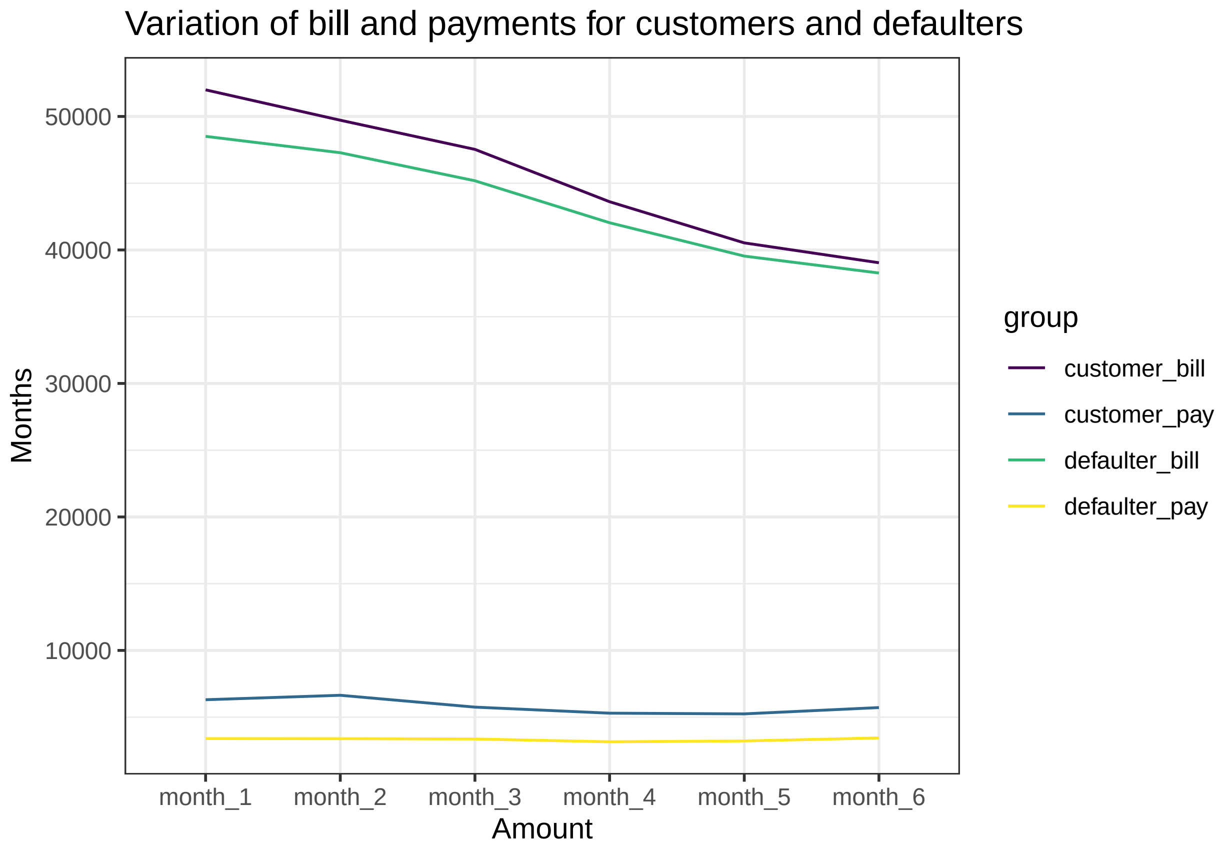
```
# Calculate the mean of payment received from customers each month
monthly_pay_cust <- df_credit_1 %>% group_by(df_pay) %>%
  summarise(month_1 = mean(payment_september),
            month_2 = mean(payment_august),
            month_3 = mean(payment_july),
            month_4 = mean(payment_june),
            month_5 = mean(payment_may),
            month_6 = mean(payment_april))

monthly_pay_cust <- as.data.frame((t(monthly_pay_cust)))

monthly_pay_cust
```

```
##               V1       V2
## df_pay         0        1
## month_1 6307.337 3397.044
## month_2 6640.465 3388.650
## month_3 5753.497 3367.352
## month_4 5300.529 3155.627
## month_5  5248.22  3219.14
## month_6 5719.372 3441.482
```

```
names(monthly_pay_cust) <- monthly_pay_cust[1,]
monthly_pay_cust <- monthly_pay_cust[-1,]

monthly_pay_cust <-monthly_pay_cust %>%
                      rename(
                        customer_pay = '0'  ,
                        defaulter_pay = '1'
                        )


# combine both bill & payment data
compare_cust <- cbind(monthly_bill_cust, monthly_pay_cust)

sapply(compare_cust, class)
```

```
##   customer_bill defaulter_bill   customer_pay  defaulter_pay
##     "character"    "character"    "character"    "character"
```

```
compare_cust$customer_bill <- as.numeric(compare_cust$customer_bill)
compare_cust$defaulter_bill <- as.numeric(compare_cust$defaulter_bill)
compare_cust$customer_pay <- as.numeric(compare_cust$customer_pay)
compare_cust$defaulter_pay <- as.numeric(compare_cust$defaulter_pay)

compare_cust$Month <- rownames(compare_cust)


data_ggp <- data.frame(x = compare_cust$Month,
                       y = c(compare_cust$customer_bill,compare_cust$defaulter_bill,
                           compare_cust$customer_pay, compare_cust$defaulter_pay),
                                 rep("defaulter_bill", nrow(compare_cust)),
                                 rep("customer_pay", nrow(compare_cust)),
                                 rep("defaulter_pay", nrow(compare_cust))))

data_ggp %>%
  ggplot( aes(x=x, y=y, group=group, color=group)) +
    geom_line() +
    scale_color_viridis(discrete = TRUE) +
    ggtitle("Variation of bill and payments for customers and defaulters") +
    xlab("Amount") + ylab("Months") +
    theme(
  axis.title.x = element_text(hjust=0.5),
  axis.title.y = element_text(hjust=0.5)
)
```



Findings : For both the bill and payments across the 6 months, the average values are significantly higher for the customers who pay their bills on time. This could mean that timely payments is an important factor in predicting the default customers

Conclusion : There are multiple factors influencing the prediction os a default customer. The history of payments is an important factor in predicting the default customers. Outstanding Debt or payments that haven't been paid is also a significant factor for finding default customers. Predicting default customers is an important task that banks do as they can save a lot of money if the customers are predicted properly. Banks can either follow up with these customers or put a cap on their credit limit to reduce the losses for banks