Test 1

1) Draw a scatter plot for the data and color the observations depending on whether the aspplication for a credit card is accepted or not.

We draw 2 ***Scatter Point plot*** to observe the correlation between variables.

```
setwd('C:/Users/啦啦露/Desktop/test 1/test 1')
dat=read.csv('CreditCard.csv')

dev.new()

pairs(card~reports+age+income+share+expenditure+owner+selfemp+dependents+months+majorcards+active,panel=panel.smooth,data=dat,main="Scatter Plot",col=dat$card)
```

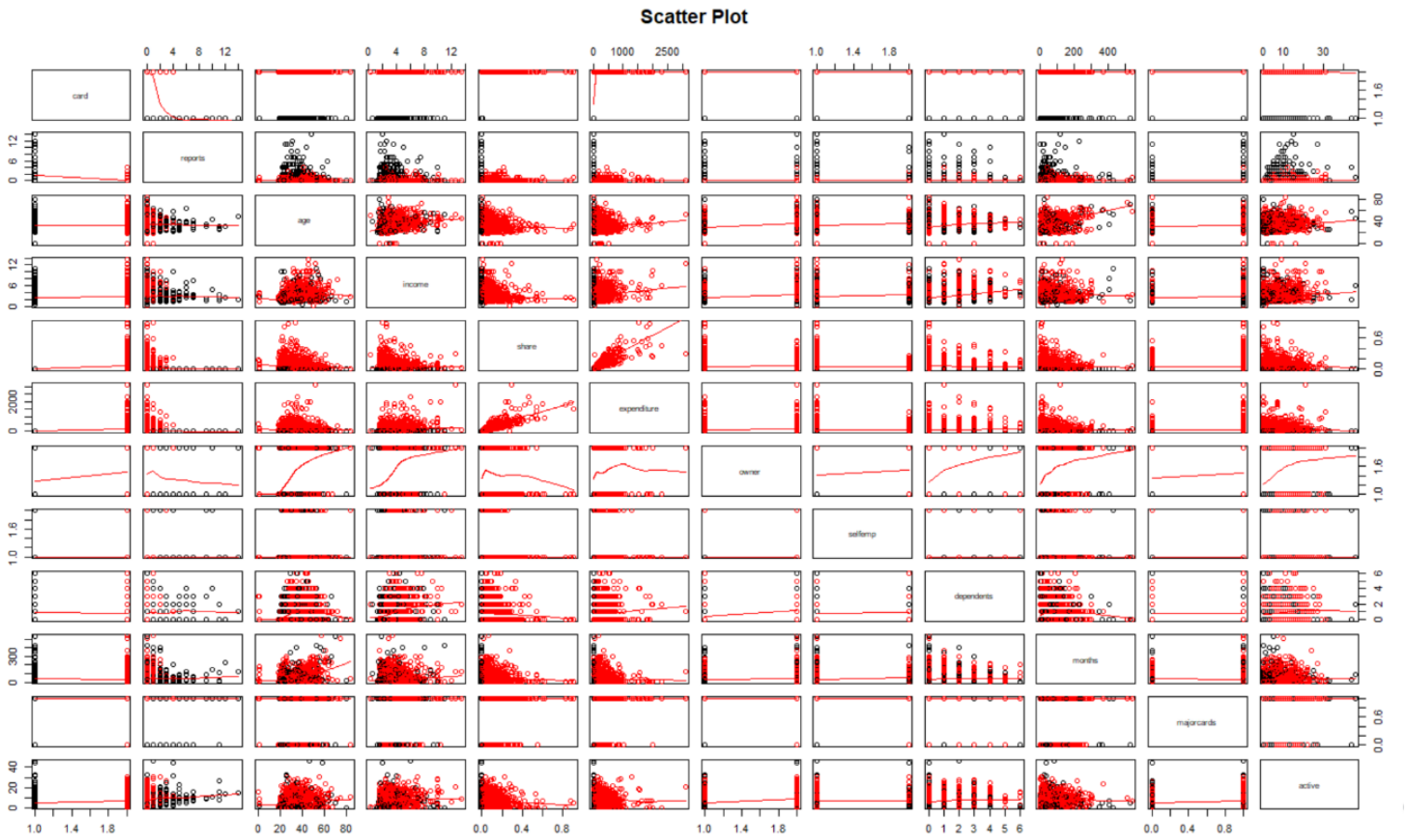

Figure 1

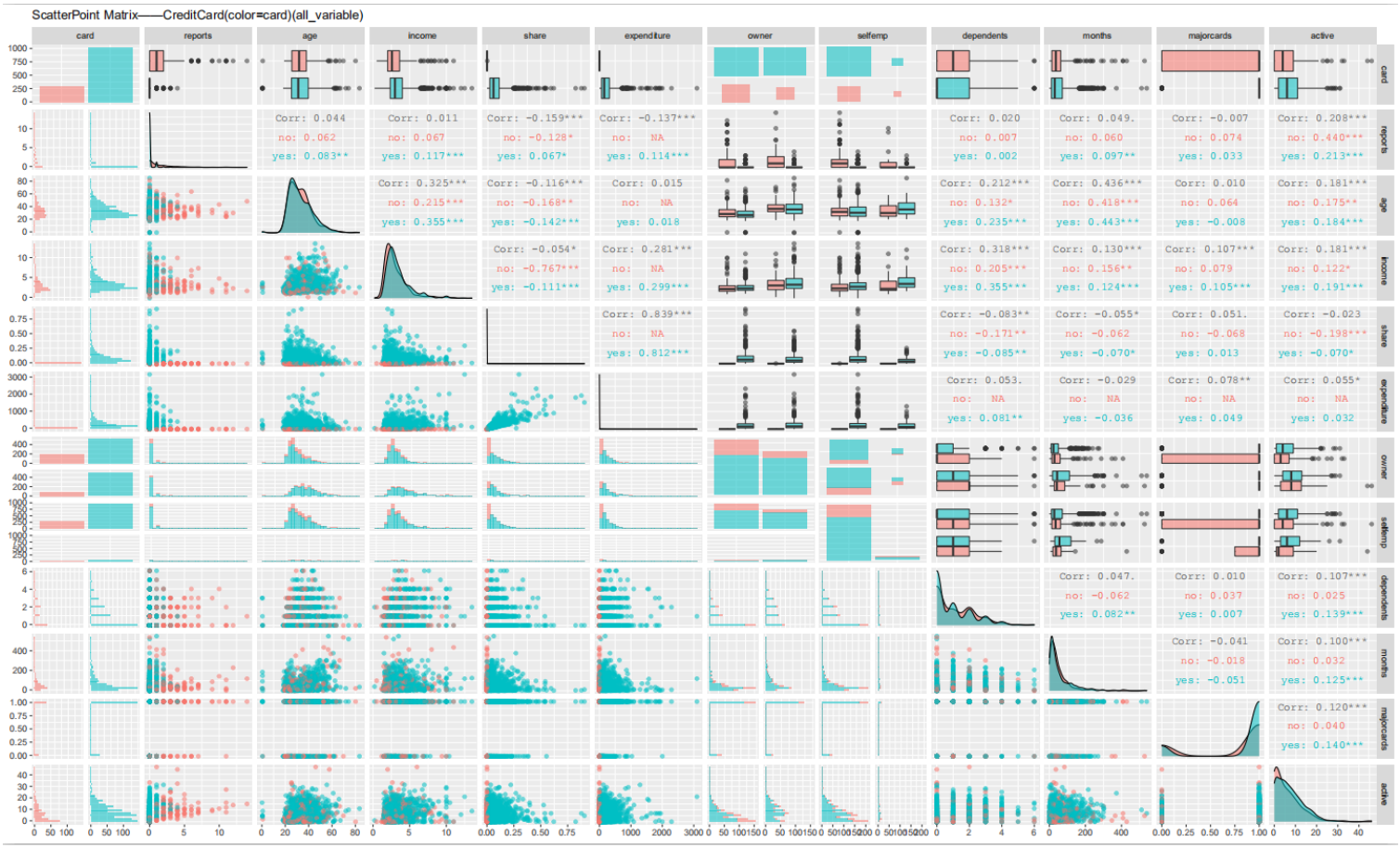

Figure 2

2) Based on the the scatter plot, pick 3-5 variables as predictors to run a logistic regression with card being the response variable.

Firstly,based on the Figure 1 and Figure 2 , we find that variables of reports, share, expenditure, month and active make difference to the response variable of card. However, we find there is a high correlation between expenditure and share, which is as high as 0.839. Therefore, we choose to delete one of them in case of multiple collinearity. As a result, we choose 4 variables : reports, share, month and active.

Secondly, we compare three different methods to fit these variables. We classify data into 2 data sets based on whether the application for a credit card accepted or not. Considering different variances in four variables in Table 1, we choose the QDA method.

Var	reports	share	month	active
YES	0.1732	0.0098	4187.376	36.9956
NO	5.8295	4.541067e-08	5115.54	48.3360

TABLE 3

```
library(MASS)
a=dat[dat$card=='yes',]
b=dat[dat$card=='no',]
var(a$active) #variance in 'yes' data set

## [1] 36.99563

var(b$active) #variance in 'no' data set

## [1] 48.33605

qda.fit=qda(card~reports+share+active+months,data=dat)
```

Lastly, we fit a ***logistic regression*** model I in order to predict ***card*** using 4 predictions, ***reports, share, active and months.*** We create a vector of 1319 ***no*** elements. Given these predictions, the table() function produces a confusion matrix in order to determine how many observations were correctly or incorrectly classified.

Hence it looks like that our model correctly predicted that the applications would be refused on 294 times and would be accepted on 1000 times. However, this result is **misleading** because we trained and tested the model on the same set.

For that reason, we quit the ***Logistic Regression***, using method QDA instead.

```
# Logistic Regression
glm.fit=glm(as.factor(card)~reports+share+active+months,data=dat,fam
ily=binomial)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

summary(glm.fit)

##
## Call:
## glm(formula = as.factor(card) ~ reports + share + active + months,

##     family = binomial, data = dat)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.373   0.000   0.000   0.000   3.064
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.647e+00  4.504e-01  -8.097 5.65e-16 ***
## reports     -2.553e+00  9.608e-01  -2.657 0.007880 **
## share        2.600e+03  4.749e+02   5.475 4.37e-08 ***
## active       1.064e-01  3.097e-02   3.435 0.000592 ***
```

```
## months       3.315e-04  3.097e-03   0.107 0.914764
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1404.6  on 1318  degrees of freedom
## Residual deviance:  148.1  on 1314  degrees of freedom
## AIC: 158.1
##
## Number of Fisher Scoring iterations: 15
```

```
glm.probs=predict(glm.fit,type='response')
glm.probs[1:10]
```

```
##         1         2         3         4         5         6         7
       8
## 1.0000000 0.9999878 0.9995514 1.0000000 1.0000000 1.0000000 1.0000
000 1.0000000
##         9        10
## 1.0000000 1.0000000
```

```
glm.pred=rep('no',1319)
glm.pred[glm.probs>0.5]='yes'
table(glm.pred,dat$card)
```

```
##
## glm.pred   no  yes
##      no   294   23
##      yes    2 1000
```

```
(294+1000)/1319
```

```
## [1] 0.9810462
```

```
mean(glm.pred==dat$card)
```

```
## [1] 0.9810462
```

3)Summarize your results and draw a confusion matrix to show the performance of your classifier.

qda.class	NO	YES
NO	295	23
YES	1	1000
TOTAL	296	1023

TABLE 4

From this model , we can find that the probability of correct prediction of the model is 98.18%, which means that our model has a good accuracy. We can see ***False Positive Rate*** and ***False Negative Rate*** are 0.34% and 2.25% respectively. Considering our dimensions, our ***False Negative Rate*** is low. Therefore, we accept our fitting result.

```
#### QDA
qda.fit=qda(card~reports+share+active+months,data=dat)
qda.fit

## Call:
## qda(card ~ reports + share + active + months, data = dat)
##
## Prior probabilities of groups:
##        no       yes
## 0.2244124 0.7755876
##
## Group means:
##       reports        share   active   months
## no  1.5878378 0.0004767955 6.054054 55.30068
## yes 0.1290323 0.0884815297 7.269795 55.25806

qda.class=predict(qda.fit,dat)$class
table(qda.class,dat$card)

##
## qda.class   no  yes
##       no   295   23
##       yes    1 1000

mean(qda.class==dat$card)

## [1] 0.9818044
```

Code Parts (code resource:test1.R)