In [1]:

```
# Loading R packages that are needed for some calculations below
install.packages("ResourceSelection")
install.packages("pROC")
install.packages("rpart.plot")
```

```
Installing package into ‘/srv/rlibs’
(as ‘lib’ is unspecified)

also installing the dependency ‘pbapply’


Installing package into ‘/srv/rlibs’
(as ‘lib’ is unspecified)

also installing the dependency ‘plyr’


Installing package into ‘/srv/rlibs’
(as ‘lib’ is unspecified)
```

In [49]:

```
# Loading credit card default data set
credit_default <- read.csv(file='credit_card_default.csv', header=TRUE, sep=",")

print("data set (first 5 observations)")
head(credit_default, 5)

print("Number of columns")
ncol(credit_default)
print("Number of rows")
nrow(credit_default)
```

```
[1] "data set (first 5 observations)"
```

A data.frame: 5 × 8

	age	sex	education	marriage	assets	missed_payment	credit_utilize	default
	<int>	<int>	<int>	<int>	<int>	<int>	<dbl>	<int>
1	28	2	2	2	0	1	0.174	0
2	25	1	1	1	1	1	1.000	1
3	49	2	1	1	0	1	0.540	1
4	26	2	2	2	3	0	0.347	0
5	38	1	1	2	2	1	0.312	0

```
[1] "Number of columns"
```

8

```
[1] "Number of rows"
```

600

In [42]:

```
# Converting appropriate variables to factors
credit_default <- within(credit_default, {
   default <- factor(default)
   sex <- factor(sex)
   education <- factor(education)
   marriage <- factor(marriage)
   assets <- factor(assets)
   missed_payment <- factor(missed_payment)
})

head(credit_default, 5)
```

A data.frame: 5 × 8

	age	sex	education	marriage	assets	missed_payment	credit_utilize	default
	<int>	<fct>	<fct>	<fct>	<fct>	<fct>	<dbl>	<fct>
1	28	2	2	2	0	1	0.174	0
2	25	1	1	1	1	1	1.000	1
3	49	2	1	1	0	1	0.540	1
4	26	2	2	2	3	0	0.347	0
5	38	1	1	2	2	1	0.312	0

In [50]:

```
# Partition the data set into training and testing data
samp.size = floor(0.70*nrow(credit_default))

# Training set
print("Number of rows for the training set")
train_ind = sample(seq_len(nrow(credit_default)), size = samp.size)
train.data1 = credit_default[train_ind,]
nrow(train.data1)

# Testing set
print("Number of rows for the validation set")
```

```
test.data1 = credit_default[-train_ind,]
nrow(test.data1)
```

```
[1] "Number of rows for the training set"
420
[1] "Number of rows for the validation set"
180
```

In [44]:

```
# Create the complete model
model1 <- glm(default ~ credit_utilize + assets + missed_payment, data = credit_default, family = "binomial")

summary(model1)
```

```
Call:
glm(formula = default ~ credit_utilize + assets + missed_payment,
    family = "binomial", data = credit_default)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.50838  -0.10623   0.00001   0.05513   2.32888

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)      -9.2371     1.2320  -7.497 6.51e-14 ***
credit_utilize   32.2826     3.9957   8.079 6.51e-16 ***
assets1          -0.4827     0.4999  -0.966 0.334240
assets2          -3.0334     0.6038  -5.024 5.05e-07 ***
assets3          -3.4568     0.5806  -5.954 2.61e-09 ***
missed_payment1   1.4276     0.4131   3.455 0.000549 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 827.93  on 599  degrees of freedom
Residual deviance: 171.23  on 594  degrees of freedom
AIC: 183.23

Number of Fisher Scoring iterations: 9
```

In [45]:

```
# Predict default or no_default for the data set using the model
default_model_data <- credit_default[c('credit_utilize', 'assets', 'missed_payment')]
pred <- predict(model1, newdata=default_model_data, type='response')

# If the predicted probability of default is >=0.50 then predict credit default (default='1'), otherwise predict no credit
# default (default='0')
depvar_pred = as.factor(ifelse(pred >= 0.5, '1', '0'))

# confusion matrix
conf.matrix <- table(credit_default$default, depvar_pred)[c('0','1'),c('0','1')]
rownames(conf.matrix) <- paste("Actual", rownames(conf.matrix), sep = ": default=")
colnames(conf.matrix) <- paste("Prediction", colnames(conf.matrix), sep = ": default=")

# confusion matrix
print("Confusion Matrix")
format(conf.matrix,justify="centre",digit=2)
```

```
[1] "Confusion Matrix"
```

A matrix: 2 × 2 of type chr

	Prediction: default=0	Prediction: default=1
Actual: default=0	262	14
Actual: default=1	21	303

In [46]:

```
library(ResourceSelection)


print("Hosmer-Lemeshow Goodness of Fit Test")
hl = hoslem.test(model1$y, fitted(model1), g=50)
hl
```

```
[1] "Hosmer-Lemeshow Goodness of Fit Test"
        Hosmer and Lemeshow goodness of fit (GOF) test

data:  model1$y, fitted(model1)
X-squared = 26.733, df = 48, p-value = 0.9945
```

In [52]:

```
print("The Hosmer-Lemeshow test results with a high p-value of 0.9945 suggest that the logistic regression model fits the data well")
```

```
[1] "The Hosmer-Lemeshow test results with a high p-value of 0.9945 suggest that the logistic regression model fits the data well"
```

In [47]:

```
library(pROC)

labels <- credit_default$default
predictions <- model1$fitted.values

roc <- roc(labels ~ predictions)

print("Area Under the Curve (AUC)")
round(auc(roc),4)

print("ROC Curve")
```

```
# True Positive Rate (Sensitivity) and False Positive Rate (1 - Specificity)
plot(roc, legacy.axes = TRUE)
```

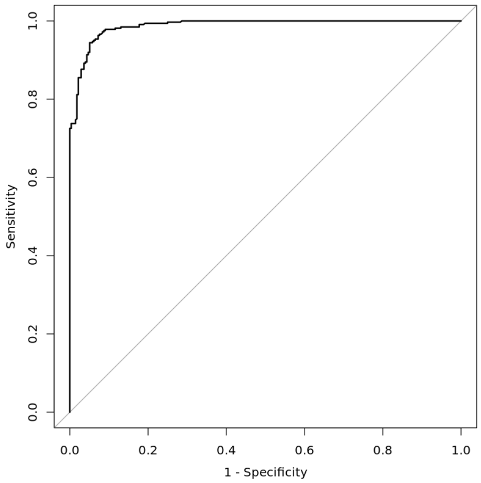
```
Setting levels: control = 0, case = 1

Setting direction: controls < cases
```

```
[1] "Area Under the Curve (AUC)"
```

0.9874

```
[1] "ROC Curve"
```



In [53]:
```
print("An AUC of 0.9874 indicates excellent model performance. It means that the model has a 98.74% chance of correctly distinguishing
```

```
[1] "An AUC of 0.9874 indicates excellent model performance. It means that the model has a 98.74% chance of correctly distinguishing be
tween a positive case (default) and a negative case (no default). This high AUC value suggests that the model is highly accurate in mak
ing predictions"
```

In [62]:
```
print("Prediction: Credit utilization: 35%, owns a car, and has missed payments in the last 3 months")
newdata1 <- data.frame(credit_utilize=0.35, assets='1', missed_payment='1')
pred1 <- predict(model1, newdata1, type='response')*100
round(pred1, 1)

print("Prediction: Credit utilization: 30%, owns a car and a house, and has not missed a payment in the last 3 months")
newdata2 <- data.frame(credit_utilize=0.30, assets='3', missed_payment='0')
pred2 <- predict(model1, newdata2, type='response')*100
round(pred2, 1)

print("Prediction: Credit utilization: 60%, owns a car and a house, and has missed a payment in the last 3 months")
newdata3 <- data.frame(credit_utilize=0.60, assets='3', missed_payment='1')
pred3 <- predict(model1, newdata3, type='response')*100
round(pred3, 1)
```

```
[1] "Prediction: Credit utilization: 35%, owns a car, and has missed payments in the last 3 months"
```

1: 95.3

```
[1] "Prediction: Credit utilization: 30%, owns a car and a house, and has not missed a payment in the last 3 months"
```

1: 4.7

```
[1] "Prediction: Credit utilization: 60%, owns a car and a house, and has missed a payment in the last 3 months"
```

1: 100