**1.1**

> sprintf("%2.3f",cumsum(lambda)/sum(lambda)\*100)

[1] "46.518" "68.590" "81.295" "91.206" "95.378" "98.489" "99.974" "100.000"

4 components needed

Scaling needed since all variables have very different scale and since PCA minimizes distance between original and projected data, the distance measure will be affected severely by variables having large scale.

**1.2**

> misclass(train$region, predict(m1))

[1] 0.05944056

> misclass(test$region, predict(m1, newdata=test))

[1] 0.07342657

Very low rates, very good model, no clear sign of ovefitting

> print(m1)

Call:

multinom(formula = region ~ ., data = train)

Coefficients:

(Intercept) PC1 PC2 PC3

2 -3.218617 -2.992163 3.187718 4.000276

3 -5.948747 -5.677377 2.827254 3.045757

Residual Deviance: 105.976

AIC: 121.976

> coef(m1)[1,]-coef(m1)[2,]

(Intercept) PC1 PC2 PC3

2.7301297 2.6852134 0.3604636 0.9545196

1/2: -3.21-2.99PC1+3.18Pc2+4PC3 =0

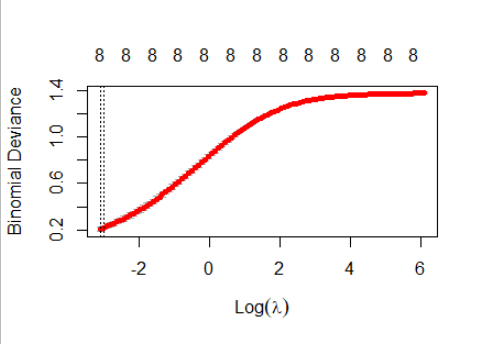
1/3: -5.94-5.67PC1+2.82Pc2+3.04PC3=0

2/3: 2.73+2.68PC1+0.36PC2+0.95PC3=0

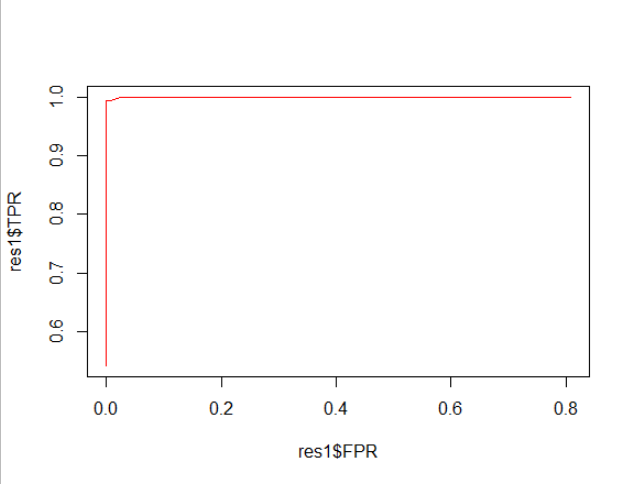
**1.3**

> model$lambda.min

[1] 0.04481765



When lambda increases, model becomes less complex and bias increases and variance decreases.



The classifier is almost perfect (close to left upper corner) and very far from random guess (45 degrees line).