FMPH243B project 1

Keren Hu

2024-01-18

```r
library(palmerpenguins)
library(mice)
library(car)
library(caret)
library(class)
library(glmnet)
library(MASS)
library(compareGroups)
library(methods)
library(tree)
library(randomForest)
library(leaps)
library(Rfast)
library(ggplot2)
library(GGally)
```

```r
dat_og = palmerpenguins::penguins
ggpairs(dat_og[,-8])
```

```
## Warning: Removed 2 rows containing non-finite values (`stat_boxplot()`).
## Removed 2 rows containing non-finite values (`stat_boxplot()`).
## Removed 2 rows containing non-finite values (`stat_boxplot()`).
## Removed 2 rows containing non-finite values (`stat_boxplot()`).
## Removed 2 rows containing non-finite values (`stat_boxplot()`).
## Removed 2 rows containing non-finite values (`stat_boxplot()`).
## Removed 2 rows containing non-finite values (`stat_boxplot()`).
## Removed 2 rows containing non-finite values (`stat_boxplot()`).
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 2 rows containing non-finite values (`stat_bin()`).
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 2 rows containing non-finite values (`stat_bin()`).
```

```
## Warning: Removed 2 rows containing non-finite values (`stat_density()`).
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 2 rows containing missing values

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 2 rows containing missing values

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 2 rows containing missing values

## Warning: Removed 2 rows containing non-finite values (`stat_boxplot()`).

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 2 rows containing non-finite values (`stat_bin()`).

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 2 rows containing non-finite values (`stat_bin()`).

## Warning: Removed 2 rows containing missing values (`geom_point()`).

## Warning: Removed 2 rows containing non-finite values (`stat_density()`).

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 2 rows containing missing values

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 2 rows containing missing values

## Warning: Removed 2 rows containing non-finite values (`stat_boxplot()`).

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 2 rows containing non-finite values (`stat_bin()`).

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 2 rows containing non-finite values (`stat_bin()`).

## Warning: Removed 2 rows containing missing values (`geom_point()`).
## Removed 2 rows containing missing values (`geom_point()`).

## Warning: Removed 2 rows containing non-finite values (`stat_density()`).

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 2 rows containing missing values

## Warning: Removed 2 rows containing non-finite values (`stat_boxplot()`).
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 2 rows containing non-finite values (`stat_bin()`).
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 2 rows containing non-finite values (`stat_bin()`).
```

```
## Warning: Removed 2 rows containing missing values (`geom_point()`).
## Removed 2 rows containing missing values (`geom_point()`).
## Removed 2 rows containing missing values (`geom_point()`).
```

```
## Warning: Removed 2 rows containing non-finite values (`stat_density()`).
```

```
## Warning: Removed 2 rows containing non-finite values (`stat_boxplot()`).
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 2 rows containing non-finite values (`stat_bin()`).
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

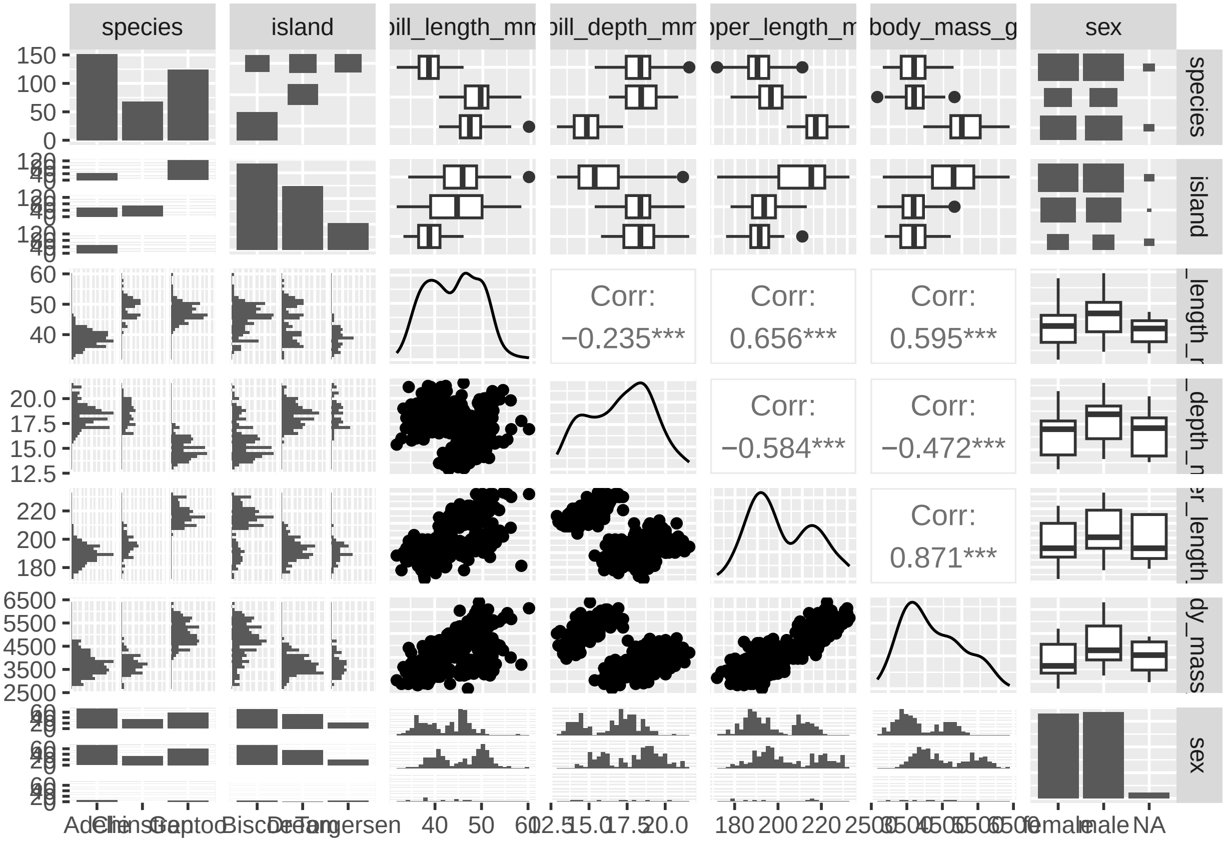
```
## Warning: Removed 2 rows containing non-finite values (`stat_bin()`).
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 2 rows containing non-finite values (`stat_bin()`).
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 2 rows containing non-finite values (`stat_bin()`).
```



```
# Demographic description
tab1 <- compareGroups(sex ~ ., data = dat_og[,-8])
restab <- createTable(tab1)
export2md(restab, caption = "Demographic characteristics of study participants")
```

Table 1: Demographic characteristics of study participants

	female N=165	male N=168	p.overall
species:			0.976
Adelie	73 (44.2%)	73 (43.5%)	
Chinstrap	34 (20.6%)	34 (20.2%)	
Gentoo	58 (35.2%)	61 (36.3%)	
island:			0.972
Biscoe	80 (48.5%)	83 (49.4%)	
Dream	61 (37.0%)	62 (36.9%)	
Torgersen	24 (14.5%)	23 (13.7%)	
bill_length_mm	42.1 (4.90)	45.9 (5.37)	<0.001
bill_depth_mm	16.4 (1.80)	17.9 (1.86)	<0.001
flipper_length_mm	197 (12.5)	205 (14.5)	<0.001
body_mass_g	3862 (666)	4546 (788)	<0.001

Missing value

```
dat = dat_og[!is.na(dat_og$sex),]

imp = mice(dat, m=5, method = "pmm", maxit = 5, seed = 2024)
```

```
##
##  iter imp variable
##   1   1
##   1   2
##   1   3
##   1   4
##   1   5
##   2   1
##   2   2
##   2   3
##   2   4
##   2   5
##   3   1
##   3   2
##   3   3
##   3   4
##   3   5
##   4   1
##   4   2
##   4   3
##   4   4
##   4   5
##   5   1
##   5   2
##   5   3
##   5   4
##   5   5
```

```
dat_imp = complete(imp, 1)
dat_imp$sex = ifelse(dat_imp$sex=="female", 0,
                     ifelse(dat_imp$sex == "male",1, NA))

table(dat_imp$sex)
```

```
##
##   0   1
## 165 168
```

Splitting dataset

```
# 70% train & 30% test
set.seed(2024)
```

```r
index = sample(1:nrow(dat_imp), 0.7*nrow(dat_imp))
train_dat = dat_imp[index, ]
test_dat = dat_imp[-index, ]
```

Logistic regression

```r
fit.log <- glm( sex ~. , data = train_dat, family = binomial)
summary(fit.log)
```

```
##
## Call:
## glm(formula = sex ~ ., family = binomial, data = train_dat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.6541  -0.0886   0.0005   0.0745   1.8652
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        1.263e+03  9.994e+02   1.264 0.206208
## speciesChinstrap  -1.135e+01  3.155e+00  -3.597 0.000322 ***
## speciesGentoo     -8.512e+00  3.910e+00  -2.177 0.029504 *
## islandDream        1.946e-01  1.183e+00   0.165 0.869336
## islandTorgersen   -4.176e-01  1.188e+00  -0.352 0.725113
## bill_length_mm     1.061e+00  2.673e-01   3.969 7.21e-05 ***
## bill_depth_mm      2.412e+00  6.544e-01   3.685 0.000228 ***
## flipper_length_mm  3.960e-03  8.149e-02   0.049 0.961244
## body_mass_g        6.344e-03  1.733e-03   3.660 0.000252 ***
## year              -6.839e-01  5.014e-01  -1.364 0.172534
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 322.659  on 232  degrees of freedom
## Residual deviance:  63.781  on 223  degrees of freedom
## AIC: 83.781
##
## Number of Fisher Scoring iterations: 8
```

```r
fit.log.step <- step(fit.log, direction = "backward")
```

```
## Start:  AIC=83.78
## sex ~ species + island + bill_length_mm + bill_depth_mm + flipper_length_mm +
##     body_mass_g + year
##
##                     Df Deviance     AIC
## - island             2   64.046  80.046
## - flipper_length_mm  1   63.783  81.783
## - year               1   65.763  83.763
## <none>                   63.781  83.781
## - species            2   86.801 102.801
## - body_mass_g        1   89.816 107.816
## - bill_depth_mm      1   90.017 108.017
## - bill_length_mm     1   94.670 112.670
##
## Step:  AIC=80.05
## sex ~ species + bill_length_mm + bill_depth_mm + flipper_length_mm +
##     body_mass_g + year
##
##                     Df Deviance     AIC
## - flipper_length_mm  1   64.046  78.046
## - year               1   65.831  79.831
## <none>                   64.046  80.046
## - species            2   89.829 101.829
## - body_mass_g        1   90.864 104.864
## - bill_depth_mm      1   91.689 105.689
## - bill_length_mm     1   94.961 108.961
##
## Step:  AIC=78.05
## sex ~ species + bill_length_mm + bill_depth_mm + body_mass_g +
##     year
##
##                  Df Deviance     AIC
## - year            1   65.860  77.860
## <none>                64.046  78.046
## - species         2   90.395 100.395
## - bill_depth_mm   1   93.039 105.039
## - bill_length_mm  1   95.596 107.596
## - body_mass_g     1   99.522 111.522
##
## Step:  AIC=77.86
## sex ~ species + bill_length_mm + bill_depth_mm + body_mass_g
##
##                  Df Deviance     AIC
## <none>                65.860  77.860
## - species         2   90.459  98.459
## - bill_depth_mm   1   95.794 105.794
## - bill_length_mm  1   95.904 105.904
## - body_mass_g     1  101.150 111.150
```

```
summary(fit.log.step)
```

```
##
## Call:
## glm(formula = sex ~ species + bill_length_mm + bill_depth_mm +
##     body_mass_g, family = binomial, data = train_dat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.8070  -0.0889   0.0006   0.0760   1.8274
##
## Coefficients:
```

```
##                    Estimate Std. Error z value Pr(>|z|)    
## (Intercept)      -1.047e+02  1.891e+01  -5.534 3.12e-08 ***
## speciesChinstrap -9.985e+00  2.719e+00  -3.672  0.00024 ***
## speciesGentoo    -7.711e+00  3.457e+00  -2.230  0.02572 *  
## bill_length_mm    9.402e-01  2.330e-01   4.035 5.45e-05 ***
## bill_depth_mm     2.393e+00  6.099e-01   3.923 8.74e-05 ***
## body_mass_g       6.450e-03  1.578e-03   4.087 4.37e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 322.66  on 232  degrees of freedom
## Residual deviance:  65.86  on 227  degrees of freedom
## AIC: 77.86
## 
## Number of Fisher Scoring iterations: 8
```

```r
log.pred = predict(fit.log.step, test_dat[, -7], type = "response")
log.pred.class = factor(ifelse(log.pred>0.5, 1, 0))

(tst.conf = table(log.pred>0.5, test_dat$sex))
```

```
##        
##          0  1
##   FALSE 48  8
##   TRUE   5 39
```

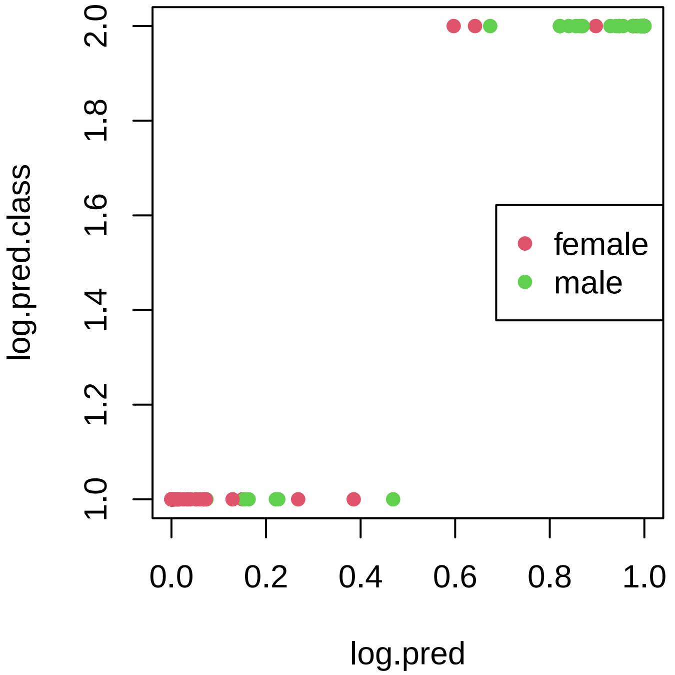
```r
(tst.error = 1 - (tst.conf[1,1] + tst.conf[2,2])/sum(tst.conf)) # 0.13
```

```
## [1] 0.13
```

```r
print(confusionMatrix(log.pred.class, as.factor(test_dat$sex)))
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction  0  1
##          0 48  8
##          1  5 39
## 
##                Accuracy : 0.87
##                  95% CI : (0.788, 0.9289)
##     No Information Rate : 0.53
##     P-Value [Acc > NIR] : 4.774e-13
## 
##                   Kappa : 0.7381
## 
##  Mcnemar's Test P-Value : 0.5791
## 
##             Sensitivity : 0.9057
##             Specificity : 0.8298
##          Pos Pred Value : 0.8571
##          Neg Pred Value : 0.8864
##              Prevalence : 0.5300
##          Detection Rate : 0.4800
##    Detection Prevalence : 0.5600
##       Balanced Accuracy : 0.8677
##
##        'Positive' Class : 0
##
```

```
par(pty="s")
plot(log.pred, log.pred.class, col=test_dat$sex+10, pch=16)
legend("right",legend = c("female", "male"),col=c(10,11), pch=16)
```



# LDA assumption: variances of all predictors are the same.

```
# fit.lda = MASS::lda(sex ~.,data = train_dat)
# lda.pred = predict(fit.lda, newdata = test_dat[,-7], type = "response")
#
# tst.conf = table(lda.pred$class, test_dat$sex)
# (tst.error = 1 - (tst.conf[1,1] + tst.conf[2,2])/sum(tst.conf)) # 0.13
#
# confusionMatrix(lda.pred$class, as.factor(test_dat$sex))
```

QDA

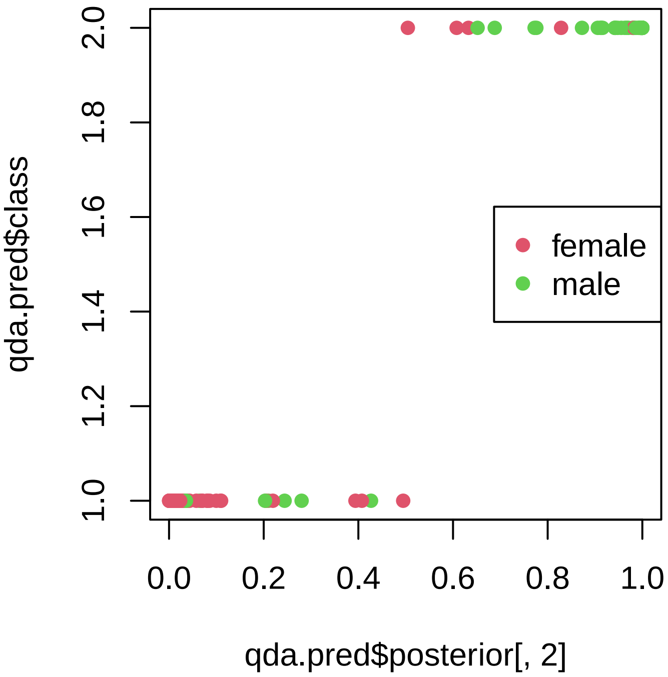
```
fit.qda = MASS::qda(sex ~.,data = train_dat)
qda.pred = predict(fit.qda, newdata = test_dat[,-7])

tst.conf = table(qda.pred$class, test_dat$sex)
(tst.error = 1 - (tst.conf[1,1] + tst.conf[2,2])/sum(tst.conf)) # 0.125
```

```
## [1] 0.12
```

```
confusionMatrix(qda.pred$class, as.factor(test_dat$sex))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 48  7
##          1  5 40
##
##                Accuracy : 0.88
##                  95% CI : (0.7998, 0.9364)
##     No Information Rate : 0.53
##     P-Value [Acc > NIR] : 7.82e-14
##
##                   Kappa : 0.7586
##
##  Mcnemar's Test P-Value : 0.7728
##
##             Sensitivity : 0.9057
##             Specificity : 0.8511
##          Pos Pred Value : 0.8727
##          Neg Pred Value : 0.8889
##              Prevalence : 0.5300
##          Detection Rate : 0.4800
##    Detection Prevalence : 0.5500
##       Balanced Accuracy : 0.8784
##
##        'Positive' Class : 0
##
```

```
par(pty="s")
plot(qda.pred$posterior[,2], qda.pred$class, col=test_dat$sex+10, pch=16)
legend("right",legend = c("female", "male"),col=c(10,11), pch=16)
```

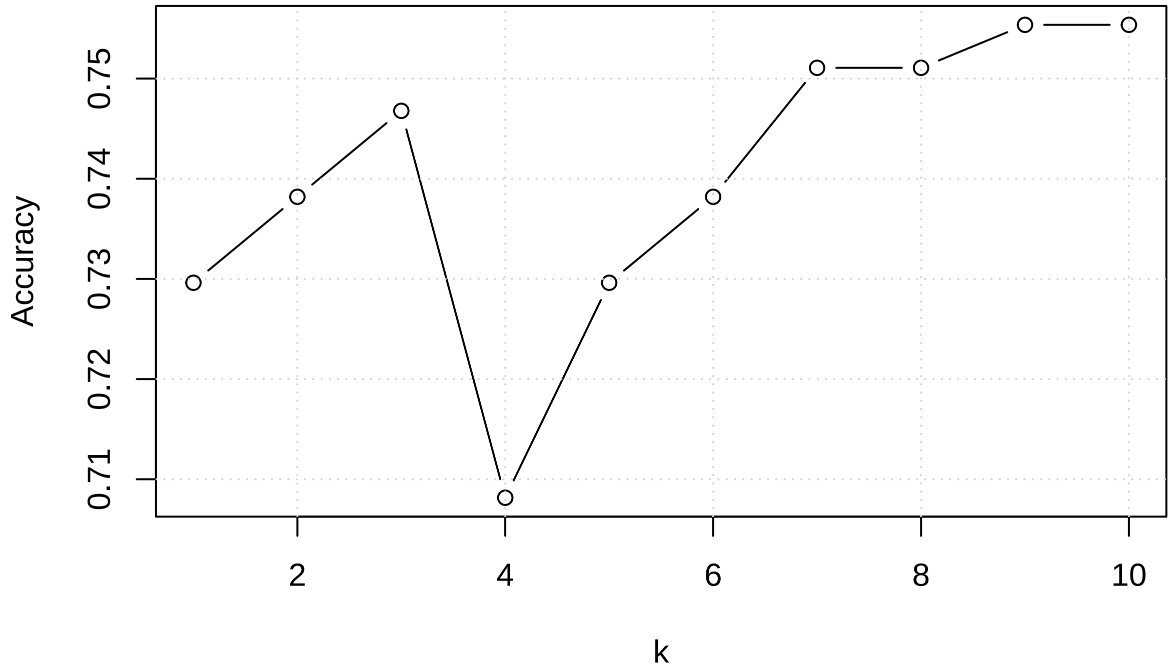


kNN

```
xxx <- as.matrix(train_dat[,-c(1,2,7)])
yyy <- train_dat[,7]

set.seed(2024)
knn.res <- Rfast::knn.cv(x=xxx, y=yyy, nfolds = 10, stratified = FALSE, k=1:10, type = "C")

(which.max(knn.res$crit)) # 9
```

```
## [1] 9
```

```
plot(1:10, knn.res$crit, type = "b", xlab = "k", ylab = "Accuracy" )
grid()
```



```r
knn.pred <- class::knn(test_dat[,-c(1,2,7)], test_dat[,-c(1,2,7)], test_dat[,7], which.max(knn.res$crit

(tst.conf = table(knn.pred, test_dat$sex))
```

```
##
## knn.pred  0  1
##        0 46 11
##        1  7 36
```

```r
(tst.error = 1 - (tst.conf[1,1] + tst.conf[2,2])/sum(tst.conf)) # 0.18
```

```
## [1] 0.18
```

```r
confusionMatrix(knn.pred, as.factor(test_dat$sex))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 46 11
##          1  7 36
##
##                Accuracy : 0.82
##                  95% CI : (0.7305, 0.8897)
##     No Information Rate : 0.53
##     P-Value [Acc > NIR] : 1.242e-09
##
##                   Kappa : 0.637
##
##  Mcnemar's Test P-Value : 0.4795
##
##             Sensitivity : 0.8679
##             Specificity : 0.7660
##          Pos Pred Value : 0.8070
##          Neg Pred Value : 0.8372
##              Prevalence : 0.5300
##          Detection Rate : 0.4600
##    Detection Prevalence : 0.5700
##       Balanced Accuracy : 0.8169
##
##        'Positive' Class : 0
##
```

Ridge

```
set.seed(2024)
gridd <- exp(seq(2,-6,-0.5))  ##lambda values

xxx <- as.matrix(train_dat[,-7])
yyy <- train_dat[,7]

##ridge fit
rdg.fit <- glmnet(xxx,yyy,family="binomial",alpha=0,lambda=gridd)
```

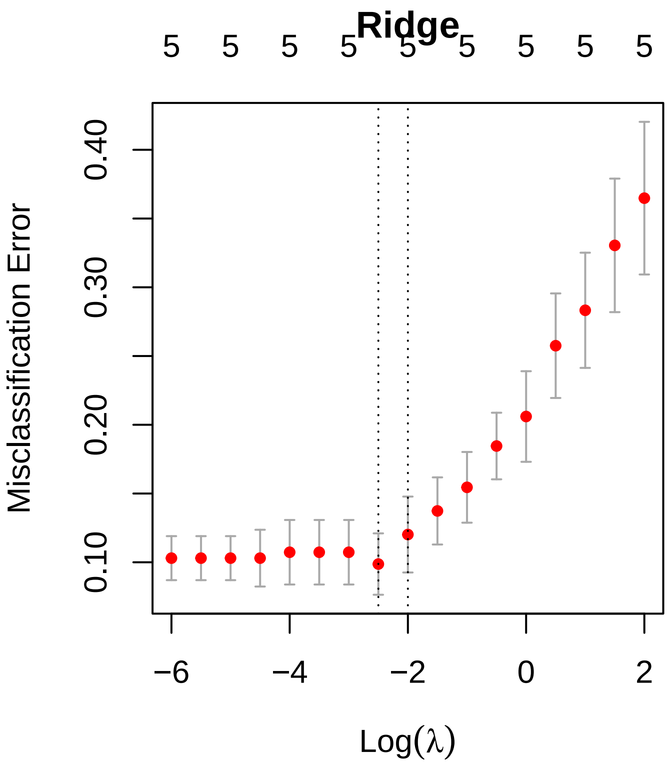
```
## Warning in storage.mode(xd) <- "double":      NA
```

```
##cross-validation to select lambda
cv.rdgeg <- cv.glmnet(xxx,yyy,family="binomial",alpha=0,
                      lambda=gridd, nfolds=10,
                      type.measure="class")
```

```
## Warning in storage.mode(xd) <- "double":      NA

## Warning in storage.mode(xd) <- "double":      NA

## Warning in storage.mode(xd) <- "double":      NA

## Warning in storage.mode(xd) <- "double":      NA

## Warning in storage.mode(xd) <- "double":      NA

## Warning in storage.mode(xd) <- "double":      NA

## Warning in storage.mode(xd) <- "double":      NA
## Warning in storage.mode(xd) <- "double":      NA

## Warning in storage.mode(xd) <- "double":      NA

## Warning in storage.mode(xd) <- "double":      NA

## Warning in storage.mode(xd) <- "double":      NA
```

```
## Warning in cbind2(1, newx) %*% nbeta:      NA

## Warning in cbind2(1, newx) %*% nbeta:      NA

## Warning in cbind2(1, newx) %*% nbeta:      NA

## Warning in cbind2(1, newx) %*% nbeta:      NA

## Warning in cbind2(1, newx) %*% nbeta:      NA

## Warning in cbind2(1, newx) %*% nbeta:      NA

## Warning in cbind2(1, newx) %*% nbeta:      NA

## Warning in cbind2(1, newx) %*% nbeta:      NA

## Warning in cbind2(1, newx) %*% nbeta:      NA

## Warning in cbind2(1, newx) %*% nbeta:      NA
```

```
par(pty="s")
plot(cv.rdgeg, main= "Ridge")
```



```
ridge.pred <- predict(cv.rdgeg, s=cv.rdgeg$lambda.1se,
                      newx = as.matrix(test_dat[,-7]),
                      type = "response")
```

```
## Warning in cbind2(1, newx) %*% nbeta:       NA
```

```
ridge.pred.class = factor(ifelse(ridge.pred>0.5, 1, 0))

(tst.conf = table(ridge.pred>0.5, test_dat$sex))
```

```
##
##          0  1
##   FALSE 46  9
##   TRUE   7 38
```

```
(tst.error = 1 - (tst.conf[1,1] + tst.conf[2,2])/sum(tst.conf)) # 0.08
```

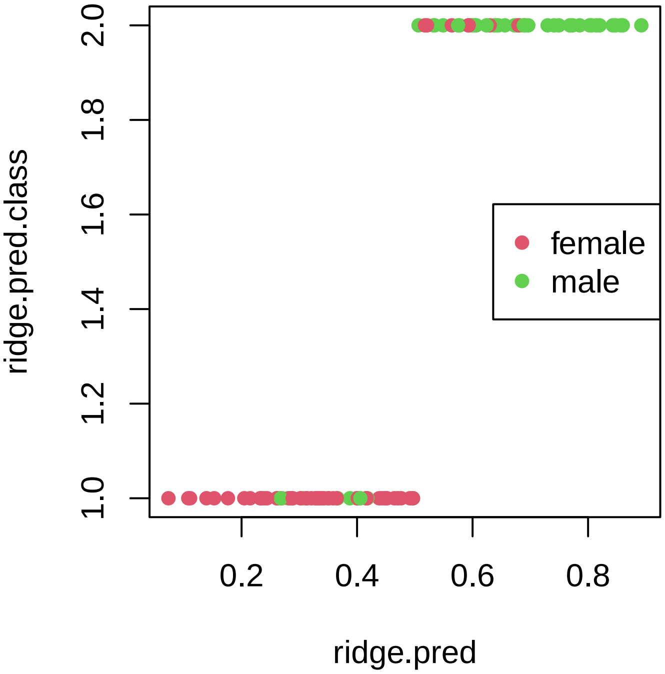
```
## [1] 0.16
```

```
confusionMatrix(ridge.pred.class, as.factor(test_dat$sex))
```

```
## Confusion Matrix and Statistics
##
```

```
##           Reference
## Prediction  0  1
##          0 46  9
##          1  7 38
##
##                Accuracy : 0.84
##                  95% CI : (0.7532, 0.9057)
##     No Information Rate : 0.53
##     P-Value [Acc > NIR] : 6.655e-11
##
##                   Kappa : 0.6781
##
##  Mcnemar's Test P-Value : 0.8026
##
##             Sensitivity : 0.8679
##             Specificity : 0.8085
##          Pos Pred Value : 0.8364
##          Neg Pred Value : 0.8444
##              Prevalence : 0.5300
##          Detection Rate : 0.4600
##    Detection Prevalence : 0.5500
##       Balanced Accuracy : 0.8382
##
##        'Positive' Class : 0
##
```

```r
par(pty="s")
plot(ridge.pred, ridge.pred.class, col=test_dat$sex+10, pch=16)
legend("right",legend = c("female", "male"),col=c(10,11), pch=16)
```

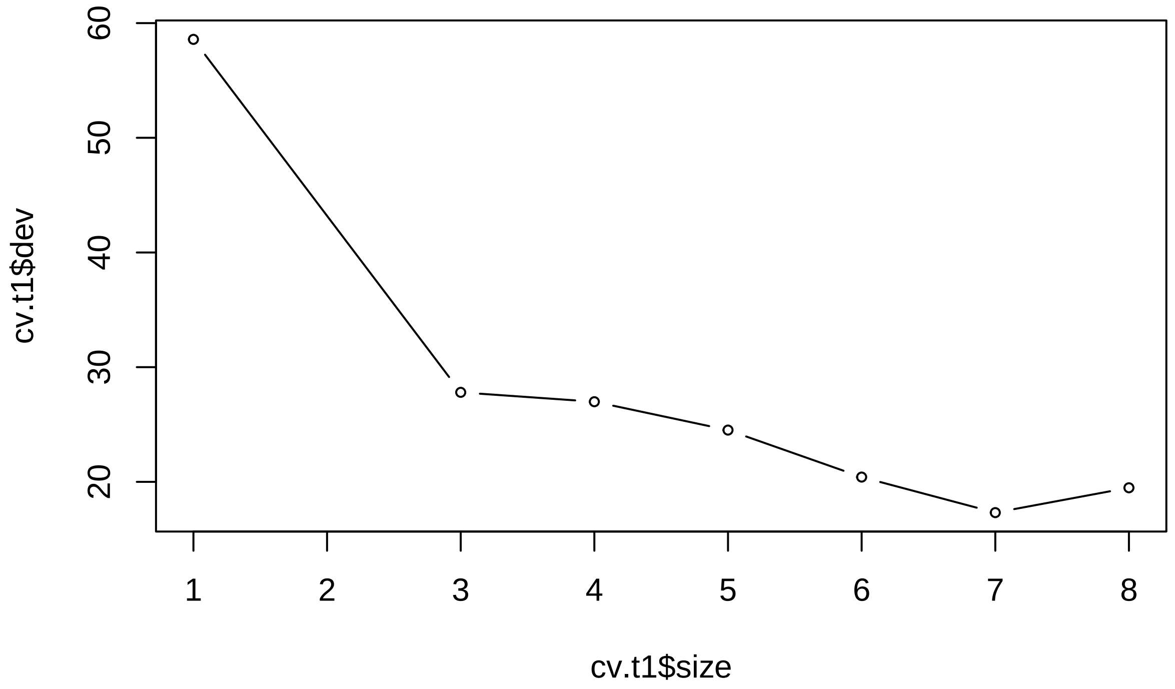


Classfication tree

```
library(tree)
set.seed(2024)
t1 <- tree(sex~., data = train_dat)
summary(t1)
```

```
##
## Regression tree:
## tree(formula = sex ~ ., data = train_dat)
## Variables actually used in tree construction:
## [1] "body_mass_g"    "bill_depth_mm"  "bill_length_mm"
## Number of terminal nodes:  8
## Residual mean deviance:  0.04305 = 9.687 / 225
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.96770  0.00000  0.03226  0.00000  0.03226  0.85710
```

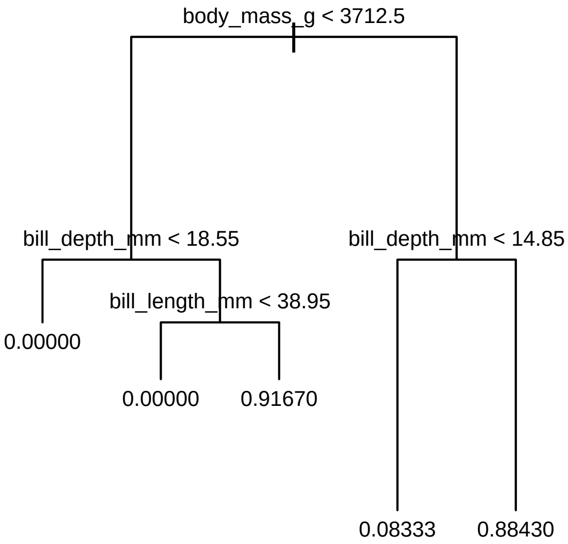
```
cv.t1 <- cv.tree(t1)
plot(cv.t1$size,cv.t1$dev,type="b",cex=0.63)
```



```
cv.t1$size[which(cv.t1$dev==min(cv.t1$dev))]
```

```
## [1] 7
```

```
prune.t1 = prune.tree(t1,best=5)
###plot the tree
par(pty="s")
plot(prune.t1)
text(prune.t1,pretty=0,cex=0.63)
```



```
tree.pred = predict(prune.t1, newdata = test_dat)
tree.pred.class = factor(ifelse(tree.pred>0.5, 1, 0))

tst.conf = table(tree.pred>0.5, test_dat$sex)
(tst.error = 1 - (tst.conf[1,1] + tst.conf[2,2])/sum(tst.conf)) # 0.096
```

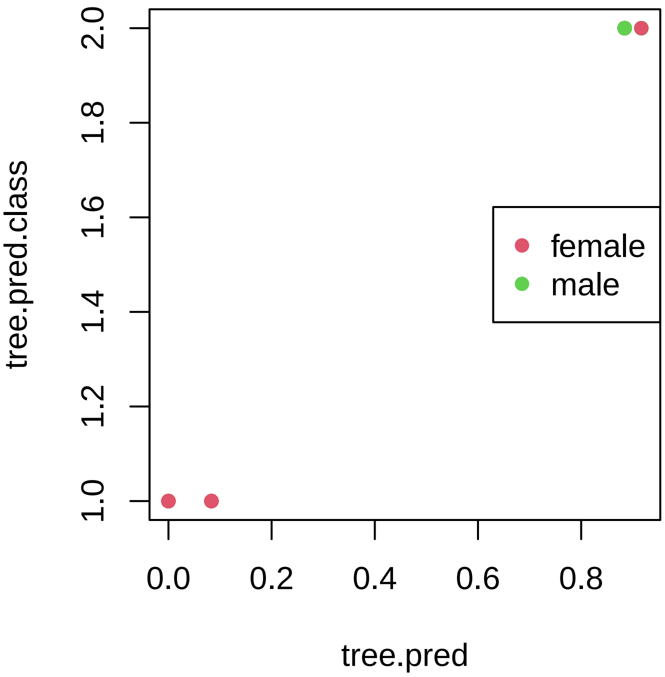
```
## [1] 0.15
```

```
confusionMatrix(tree.pred.class, as.factor(test_dat$sex))
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction  0  1
##          0 44  6
##          1  9 41
##
##                Accuracy : 0.85
##                  95% CI : (0.7647, 0.9135)
##     No Information Rate : 0.53
##     P-Value [Acc > NIR] : 1.386e-11
##
##                   Kappa : 0.7
##
##  Mcnemar's Test P-Value : 0.6056
```

```
##
##             Sensitivity : 0.8302
##             Specificity : 0.8723
##          Pos Pred Value : 0.8800
##          Neg Pred Value : 0.8200
##              Prevalence : 0.5300
##          Detection Rate : 0.4400
##    Detection Prevalence : 0.5000
##       Balanced Accuracy : 0.8513
##
##        'Positive' Class : 0
##
```

```
par(pty="s")
plot(tree.pred, tree.pred.class, col=test_dat$sex+10, pch=16)
legend("right",legend = c("female", "male"),col=c(10,11), pch=16)
```


tree.pred.class
tree.pred
● female
● male