Homework 3

ZHUOHUI LIANG

1

1

```
data("Weekly")

Weekly = Weekly %>%
  janitor::clean_names() %>%
  select(-today)

skimr::skim_without_charts(Weekly)
```

Table 1: Data summary

Name	Weekly
Number of rows	1089
Number of columns	8
Column type frequency:	
factor	1
numeric	7
Group variables	None

Variable type: factor

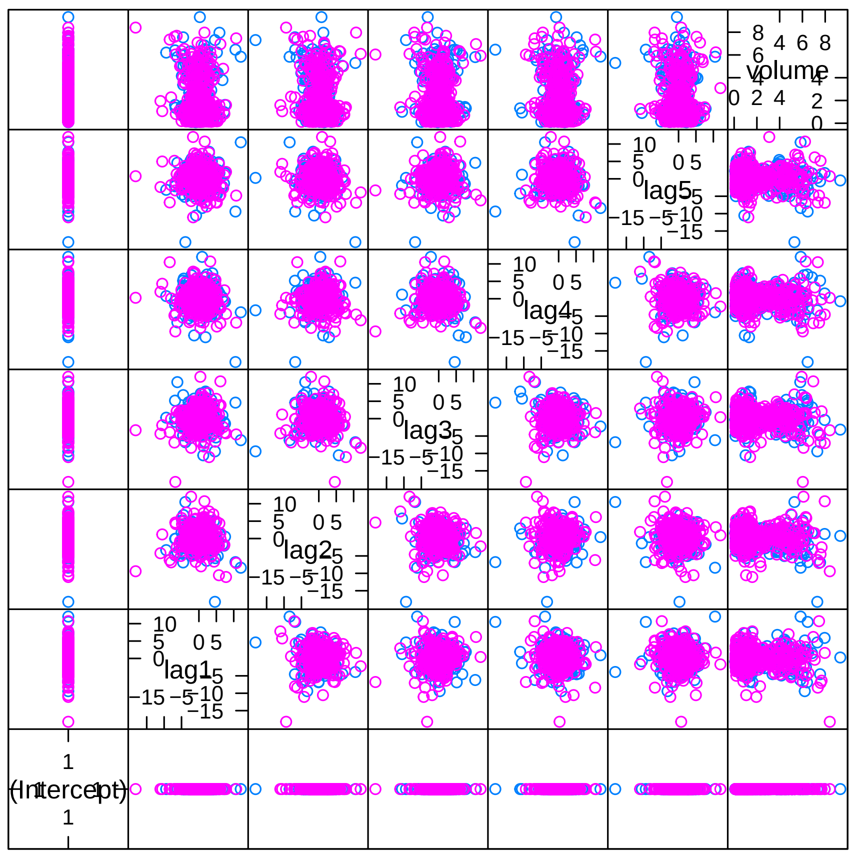
skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
direction	0	1	FALSE	2	Up: 605, Dow: 484

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
year	0	1	2000.05	6.03	1990.00	1995.00	2000.00	2005.00	2010.00
lag1	0	1	0.15	2.36	-18.20	-1.15	0.24	1.41	12.03
lag2	0	1	0.15	2.36	-18.20	-1.15	0.24	1.41	12.03
lag3	0	1	0.15	2.36	-18.20	-1.16	0.24	1.41	12.03
lag4	0	1	0.15	2.36	-18.20	-1.16	0.24	1.41	12.03

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
lag5	0	1	0.14	2.36	-18.20	-1.17	0.23	1.41	12.03
volume	0	1	1.57	1.69	0.09	0.33	1.00	2.05	9.33

```
caret::featurePlot(model.matrix(direction~lag1+lag2+lag3+lag4+lag5+volume,Weekly %>% select(-year)),Wee
```

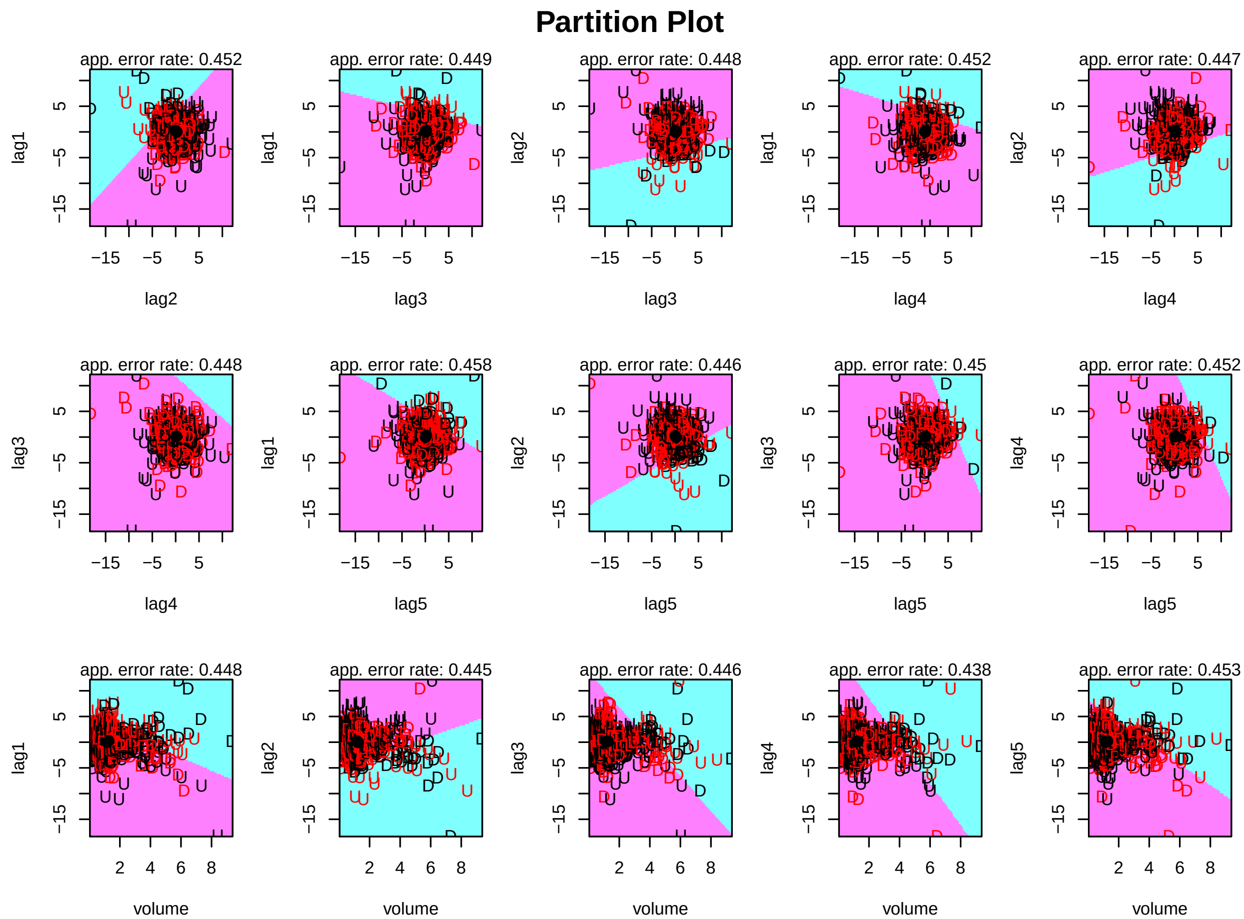


Scatter Plot Matrix

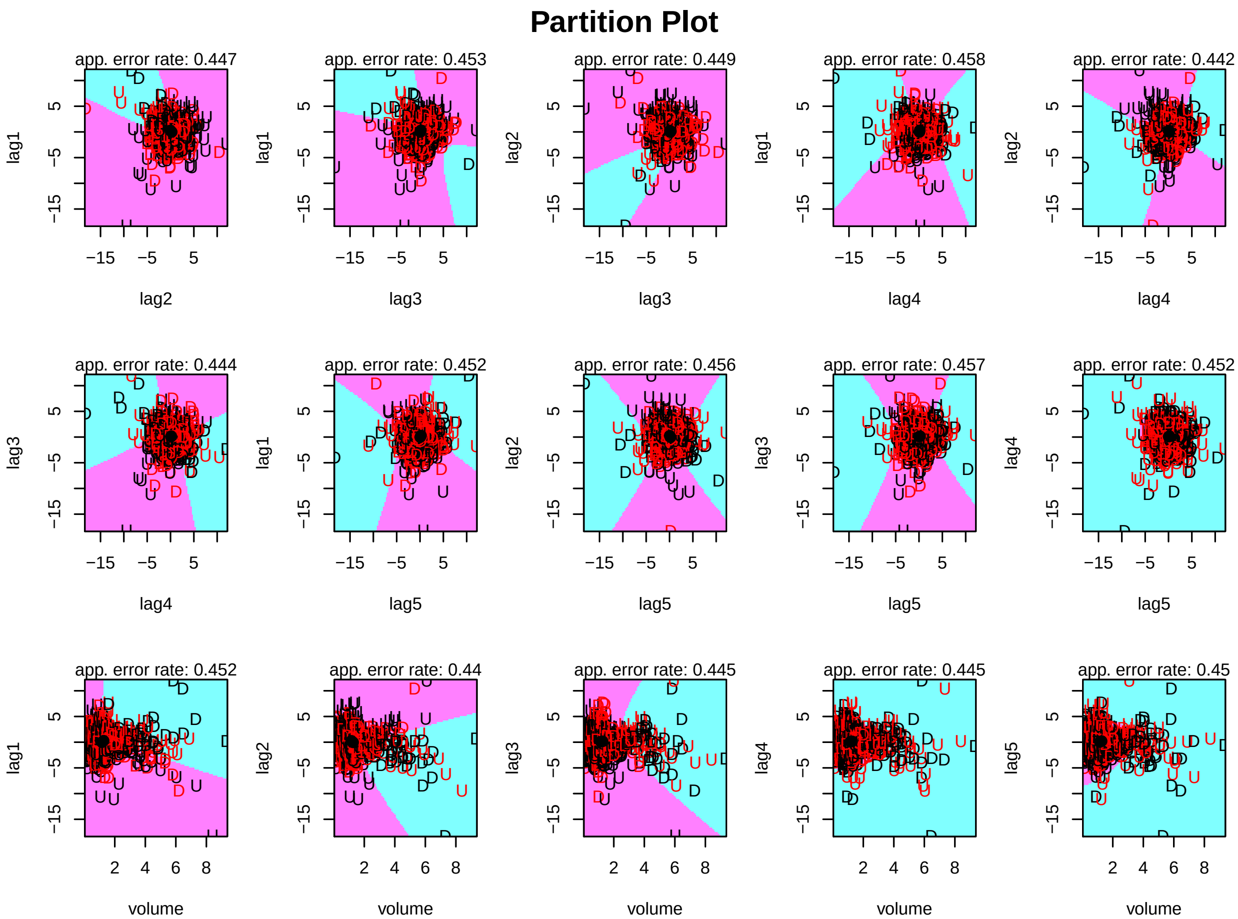
```
Weekly_Tr = Weekly %>%
  filter(year <=2008)

Weekly_Ts = Weekly %>%
  filter(year > 2008)


partimat(direction~lag1+lag2+lag3+lag4+lag5+volume,Weekly_Tr,method = "lda",nplots.vert=3,nplots.hor=5)
```




```
partimat(direction~lag1+lag2+lag3+lag4+lag5+volume,Weekly_Tr,method = "qda",nplots.vert=3,nplots.hor=5)
```



Above images has shown that there're massive overlaying in all predictors,the prediction may perform poorly.

2

```r
Weekly_logistic =
  train(
    X_tr,
    Y_tr,
    method = "glm",
    family = "binomial",
    trControl = TRC,
    metric = "ROC",
    preProcess = c("center", "scale")
  )
```

```r
logistic_prediction =
  predict(Weekly_logistic,newdata = X_ts, type = "raw")

confusionMatrix(logistic_prediction,Y_ts)
```

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

```
##           Reference
## Prediction Down Up
##       Down   31 44
##       Up     12 17
##
##                Accuracy : 0.462
##                  95% CI : (0.363, 0.562)
##     No Information Rate : 0.587
##     P-Value [Acc > NIR] : 0.996
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : 3.43e-05
##
##             Sensitivity : 0.721
##             Specificity : 0.279
##          Pos Pred Value : 0.413
##          Neg Pred Value : 0.586
##              Prevalence : 0.413
##          Detection Rate : 0.298
##    Detection Prevalence : 0.721
##       Balanced Accuracy : 0.500
##
##        'Positive' Class : Down
##
```

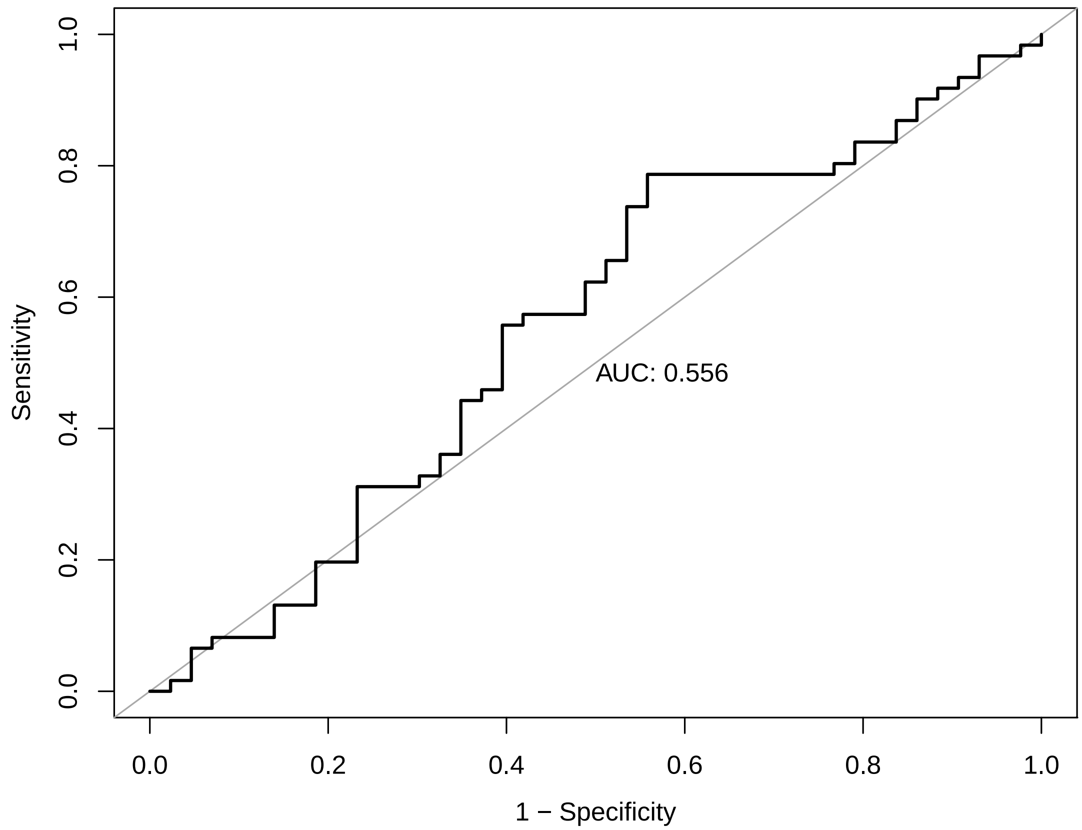
The `accuarcy` of the model is 0.462, which is worse than taking a random guessing(0.5). This conclusion can be draw by `Kappa`, which is 0. So this model perform poorly.

3

```
Weekly_logistic2 =
  train(model.matrix(direction~lag1+lag2,Weekly_Tr)[,-1],
        Y_tr,
        method = "glm",
        metric = "ROC",
        trControl = TRC,
        preProcess = c("center","scale"))
```

```
logistic_roc =
  pROC::roc(Y_ts,predict(Weekly_logistic2,newdata = X_ts,type = "prob")[,2])

ggplotify::as.ggplot(~plot(logistic_roc,legacy.axes = TRUE,,print.auc=T))
```



As shown, the model is just slightly better than guessing, with `AUC` $= 0.556$.

4-5

LDA

```
Weekly_lda =
  train(model.matrix(direction~lag1+lag2,Weekly_Tr)[,-1],
        Y_tr,
        method = "lda",
        metric = "ROC",
        trControl = TRC,
        preProcess = c("center","scale"))
```

```
lda_roc = pROC::roc(Y_ts,predict(Weekly_lda,newdata = X_ts,type = "prob")[,2])
```

QDA

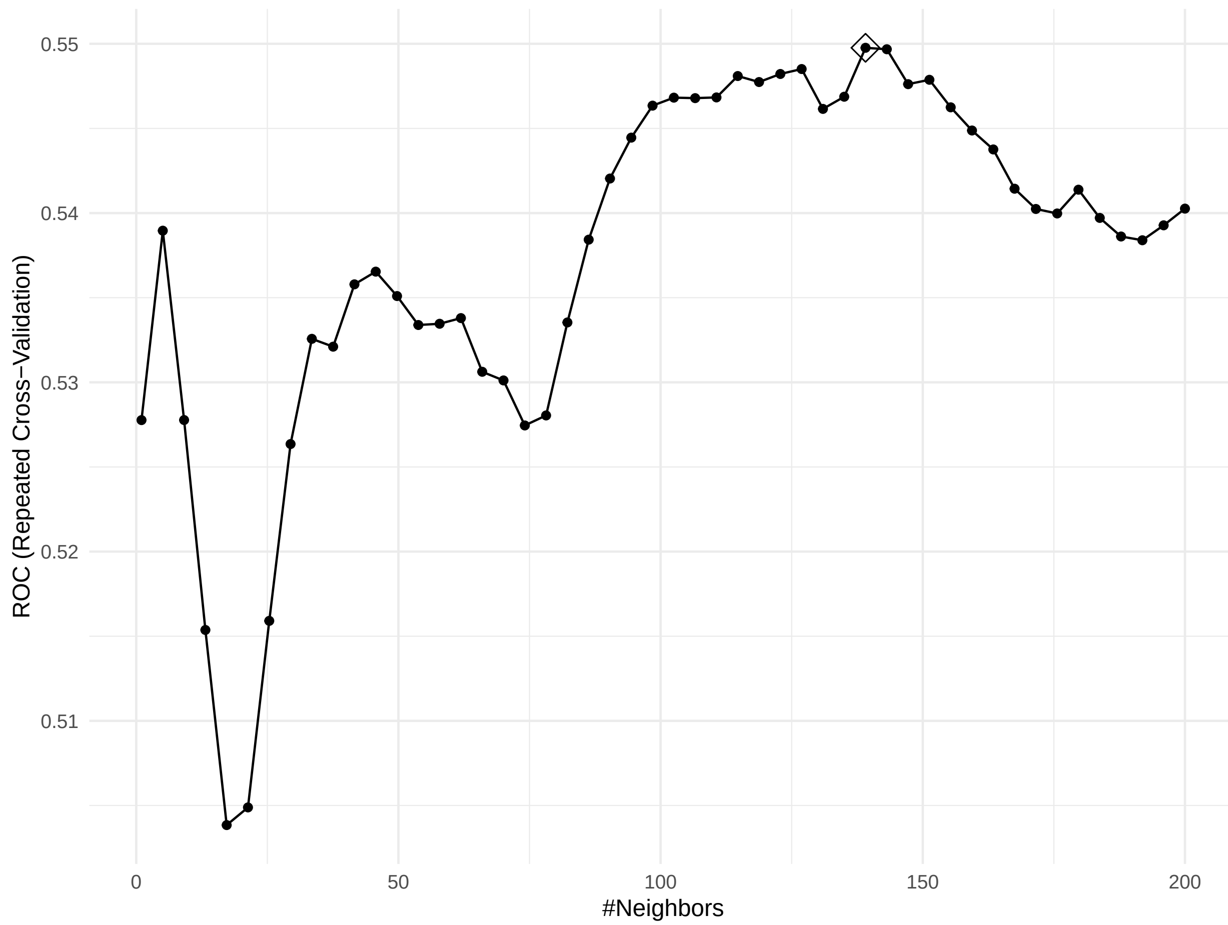
```
Weekly_qda =
  train(model.matrix(direction~lag1+lag2,Weekly_Tr)[,-1],
```

```r
        Y_tr,
        method = "qda",
        metric = "ROC",
        trControl = TRC,
        preProcess = c("center","scale"))

qda_roc = pROC::roc(Y_ts,predict(Weekly_qda,newdata = X_ts,type = "prob")[,2])
```

```r
cl = makePSOCKcluster(5) #if windows, set to 1
registerDoParallel(cl)
Weekly_knn =
  train(model.matrix(direction~lag1+lag2,Weekly_Tr)[,-1],
        Y_tr,
        method = "knn",
        metric = "ROC",
        tuneGrid = expand.grid(k=seq(1,200,len=50)),
        trControl = TRC,
        preProcess = c("center","scale"))
stopCluster(cl)

ggplot(Weekly_knn,highlight = T)
```



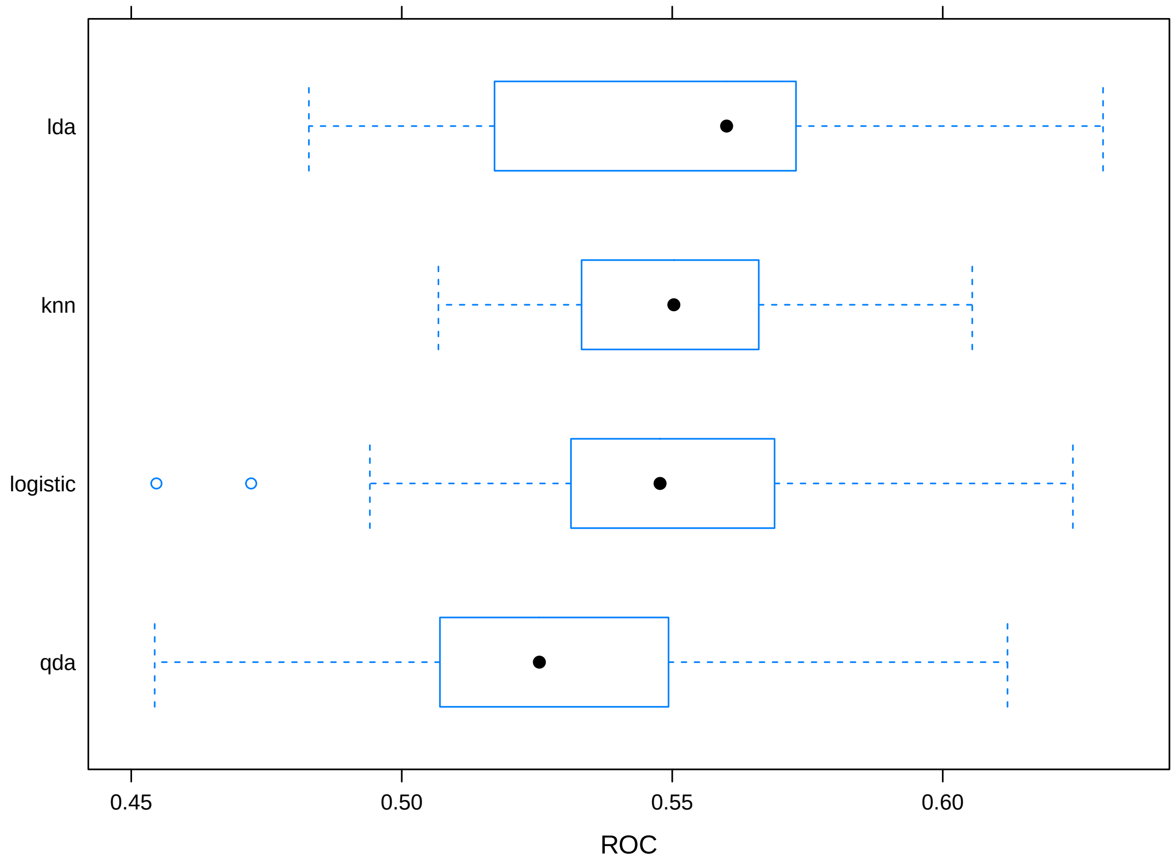
```r
knn_roc = pROC::roc(Y_ts,predict(Weekly_knn,newdata = X_ts,type = "prob")[,2])
```

```r
rsmp = resamples(list(
  logistic = Weekly_logistic2,
  lda = Weekly_lda,
  qda = Weekly_qda,
  knn = Weekly_knn
))

summary(rsmp)
```

```
##
## Call:
## summary.resamples(object = rsmp)
##
## Models: logistic, lda, qda, knn
## Number of resamples: 25
##
## ROC
##           Min. 1st Qu. Median  Mean 3rd Qu.  Max. NA's
## logistic 0.455   0.531  0.548 0.545   0.569 0.624    0
## lda      0.483   0.517  0.560 0.549   0.573 0.630    0
## qda      0.454   0.507  0.525 0.524   0.549 0.612    0
## knn      0.507   0.533  0.550 0.550   0.566 0.605    0
##
## Sens
##            Min. 1st Qu. Median   Mean 3rd Qu.  Max. NA's
## logistic 0.0455  0.0787 0.0909 0.0988   0.125 0.180    0
## lda      0.0227  0.0674 0.1023 0.0908   0.114 0.159    0
## qda      0.0000  0.1364 0.1910 0.1904   0.236 0.404    0
## knn      0.1364  0.2386 0.2614 0.2726   0.307 0.364    0
##
## Spec
##           Min. 1st Qu. Median  Mean 3rd Qu.  Max. NA's
## logistic 0.789   0.890  0.908 0.906   0.927 0.972    0
## lda      0.817   0.890  0.917 0.919   0.963 0.982    0
## qda      0.630   0.771  0.843 0.823   0.862 1.000    0
## knn      0.670   0.716  0.778 0.770   0.807 0.881    0
```

```r
bwplot(rsmp,metric = "ROC")
```



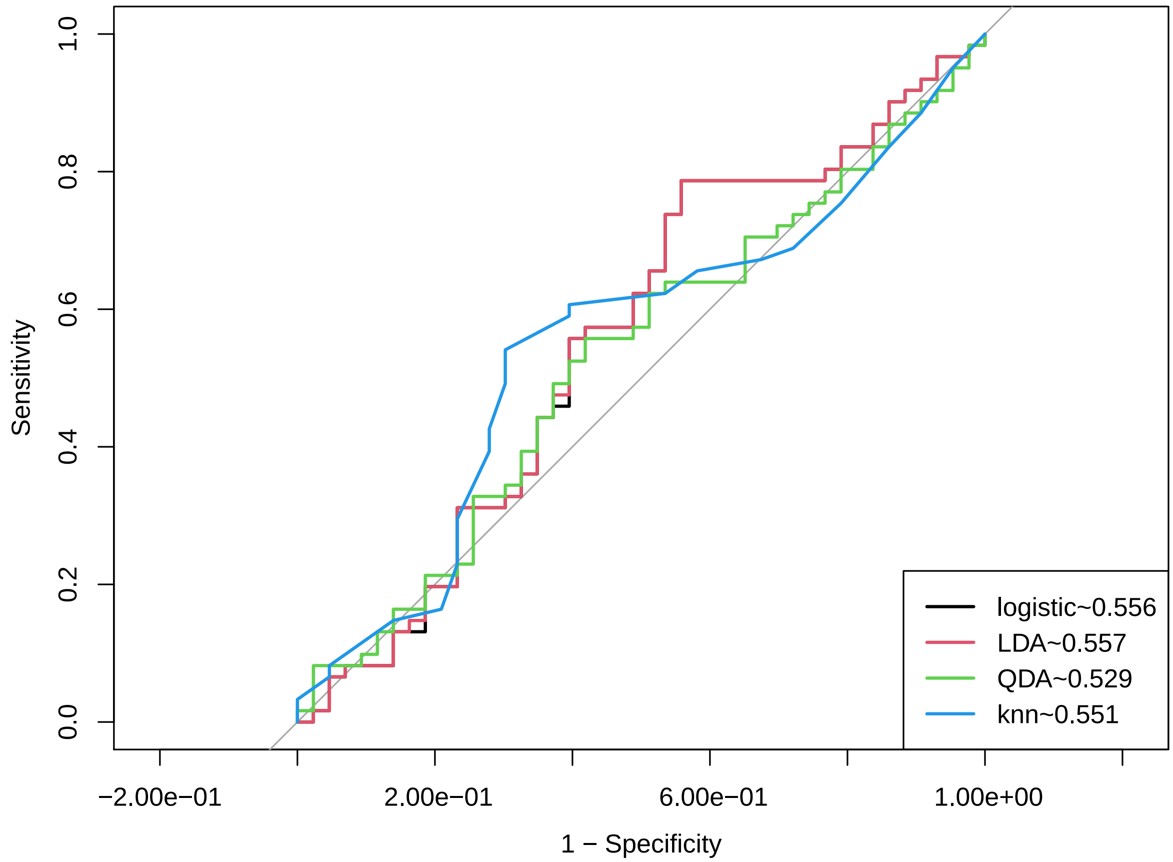

```
auc = c()

ROC = list(logistic_roc,lda_roc,qda_roc,knn_roc)

for (i in 1:4){
  auc = append(auc,ROC[[i]]$auc[1])
  plot(ROC[[i]],col = i, add = T * (i>1), legacy.axes = T * (i==1))
}

model_name =
  c("logistic","LDA","QDA","knn")

legend("bottomright",
       legend = paste0(model_name,"~",round(auc,3)),col=1:4,lwd=2)
```



With resampling all models above, none of the models has a mean `ROC` predicability above 60%, and `lda`, although has the highest mean `ROC` but has as large variance, `knn` however, has a higher median and lower variance than `lda`.

With the test data, LDA has the highest but still relatively low `AUC`, and followed by `logistic`.