Replication Code for 'Is Random Forest Really Better than Logistic Regression for Predicting Civil War Onsets?'

```
source("packages.R")
```

```
## Warning: package 'caret' was built under R version 3.3.2
```

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

```
## Warning: package 'pROC' was built under R version 3.3.2
```

The packages needed for the analysis will be automatically installed (if they aren't installed yet) and loaded.

```
registerDoMC(cores = detectCores()-1)
```

```
rm(list=ls())

data <- read.csv(file="SambnisImp.csv")
```

Data Pre-processing

```
df <- data[,c("warstds", "ager", "agexp", "anoc", "army85", "autch98", "auto4",
              "autonomy", "avgnabo", "centpol3", "coldwar", "decade1", "decade2",
              "decade3", "decade4", "dem", "dem4", "demch98", "dlang", "drel",
              "durable", "ef", "ef2", "ehet", "elfo", "elfo2", "etdo4590",
              "expgdp", "exrec", "fedpol3", "fuelexp", "gdpgrowth", "geo1", "geo2",
              "geo34", "geo57", "geo69", "geo8", "illiteracy", "incumb", "infant",
              "inst", "inst3", "life", "lmtnest", "ln_gdpen", "lpopns", "major",
              "manuexp", "milper", "mirps0", "mirps1", "mirps2", "mirps3", "nat_war", "ncontig",
              "nmgdp", "nmdp4_alt", "numlang", "nwstate", "oil", "p4mchg",
              "parcomp", "parreg", "part", "partfree", "plural", "plurrel",
              "pol4", "pol4m", "pol4sq", "polch98", "polcomp", "popdense",
              "presi", "pri", "proxregc", "ptime", "reg", "regd4_alt", "relfrac",
              "seceduc","second", "semipol3", "sip2", "sxpnew", "sxpsq",
              "tnatwar", "trade", "warhist", "xconst")]

# Converting DV into Factor with names for Caret Library

df$warstds<-factor(
  df$warstds,
  levels=c(0,1),
  labels=c("peace", "war"))
```

```
dfTrain <- df[data$year < 1990,]
dfTest <- df[data$year >= 1990,]
```

Training the models

All models are trained using the `caret` R-package.

Setting up train control

```
# create a list of seesd, here change the seed for each resampling
set.seed(221)

# length is = (n_repeats*nresampling)+1
seeds <- vector(mode = "list", length = 11)

#3 is the number of tuning parameters, mtry for rf
for(i in 1:10) seeds[[i]]<- sample.int(n=1000, 3)

#for the final model
set.seed(221)
seeds[[11]]<-sample.int(1000, 1)

# Create folds
folds <- 10
cvIndex <- createFolds(factor(dfTrain$warstds), folds, returnTrain = T)

tc <- trainControl(method = "cv",
                   number = folds,
                   summaryFunction = twoClassSummary,
                   classProb = TRUE,
                   savePredictions = TRUE,
                   seeds = seeds,
                   index = cvIndex
)
```

The models with the Fearon/Laitin variables.

First, we train the Logit and Random Forest model using the eleven Fearon/Laitin variables. And have a brief first look at the results.

Logit

```
modelFLlogit <- train(warstds ~ warhist + ln_gdpen + lpopns + lmtnest + ncontig + oil
                      + nwstate + inst3 + pol4 + ef + relfrac,
                      metric = "ROC", method = "glm", family = "binomial",
                      trControl = tc, data = dfTrain
                      )

summary(modelFLlogit) # Not in paper

modelFLlogit # Not in paper

confusionMatrix(modelFLlogit, norm = "average") # Not in paper
```

Random Forest

```
modelFLrf <- train(warstds ~ warhist + ln_gdpen + lpopns + lmtnest + ncontig + oil
                   + nwstate + inst3 + pol4 + ef + relfrac,
                   metric = "ROC", method = "rf",
                   trControl = tc, data = dfTrain,
                   keep.inbag = TRUE
                   )

modelFLrf # Not in paper

confusionMatrix(modelFLrf, norm = "average") # Not in paper
```

The models using all 90 variables in the dataset.

Logit

```
modelAlllogit <- train(warstds ~ .,
                       metric = "ROC", method = "glm", family = "binomial",
                       trControl = tc, data = dfTrain
)
```

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

```
summary(modelAlllogit) # Not in paper

modelAlllogit # Not in paper

confusionMatrix(modelAlllogit, norm = "average") # Not in paper
```

Random Forest

```
modelAllrf <- train(warstds ~ .,
                    metric = "ROC", method = "rf",
                    trControl = tc, data = dfTrain,
                    keep.inbag = TRUE
)

modelAllrf # Not in paper

confusionMatrix(modelAllrf, norm = "average") # Not in paper
```

The plots and confusion matrices to compare the models

Entries for Table 1: Confusion Matrices for the models at a threshold for positive prediction of 0.5

```
table(pred = predict(modelFLlogit, dfTest, type = "raw"), obs = dfTest$warstds)
```

```
##        obs
## pred    peace  war
##   peace  1813   28
##   war       0    0
```

```
table(pred = predict(modelFLrf, dfTest, type = "raw"), obs = dfTest$warstds)
```

```
##        obs
## pred    peace  war
##   peace  1811   25
##   war       2    3
```

```
table(pred = predict(modelAlllogit, dfTest, type = "raw"), obs = dfTest$warstds)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

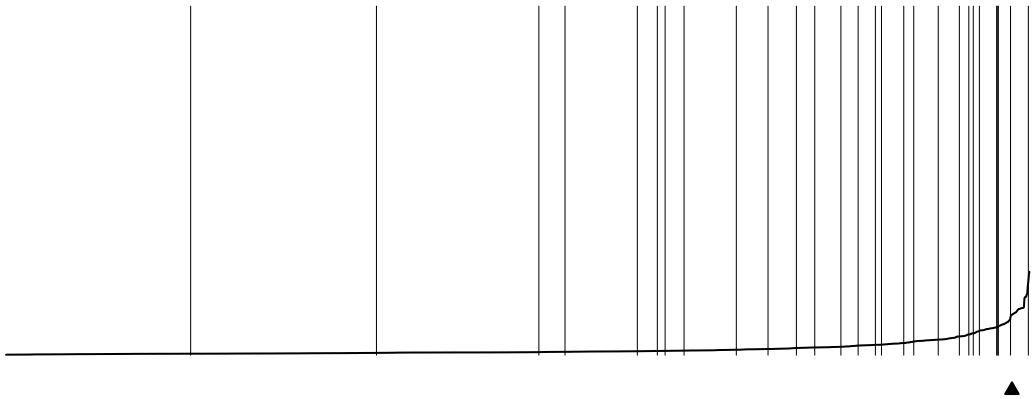
```
##        obs
## pred    peace  war
##   peace  1788   26
##   war      25    2
```

```
table(pred = predict(modelAllrf, dfTest, type = "raw"), obs = dfTest$warstds)
```

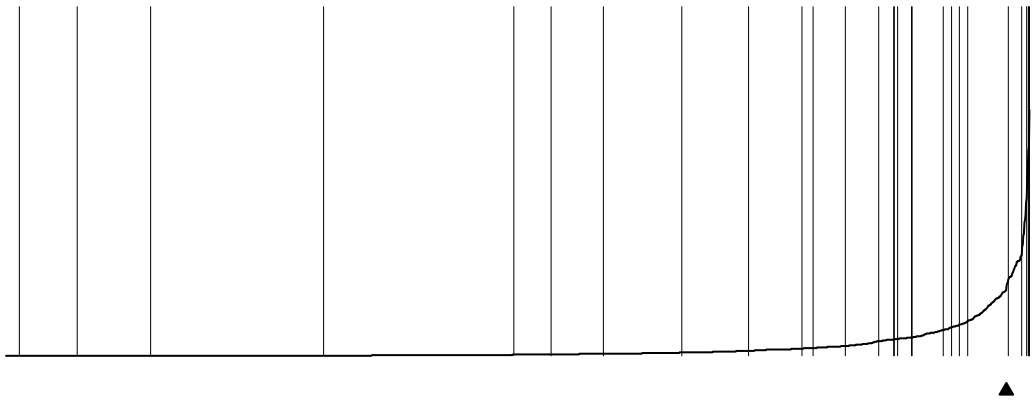
```
##        obs
## pred    peace  war
##   peace  1812   26
##   war       1    2
```

Figure 1: Separation Plots for the four models

```
separationplot(predict(modelFLlogit, dfTest, type = "prob")$war,
               as.numeric(dfTest$warstds)-1, type = "line",
               line = T, lwd2 = 1,
               show.expected = T,
               heading = "Fearon and Laitin Variables (2003) Logit",
               height = 1.5, col0 = "white", col1 = "black", newplot = F)
```

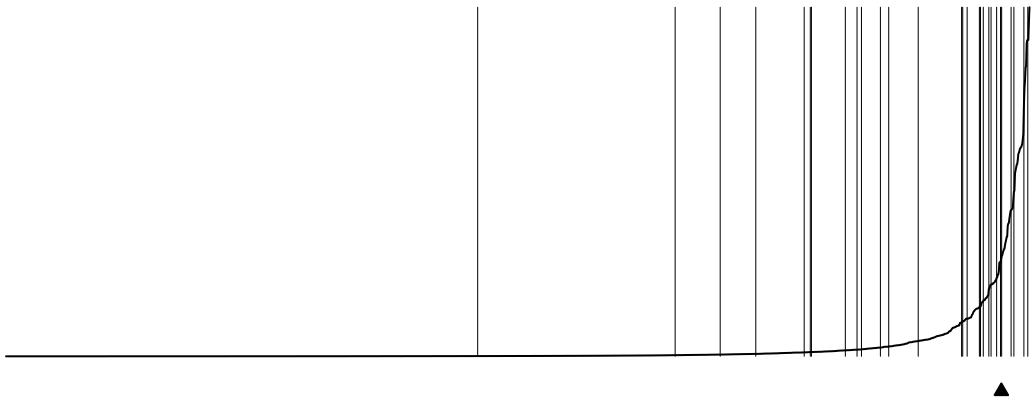


```
separationplot(predict(modelFLrf, dfTest, type = "prob")$war,
               as.numeric(dfTest$warstds)-1, type = "line",
               line = T, lwd2 = 1,
               show.expected = T,
               heading = "Fearon and Laitin Variables (2003) Random Forest",
               height = 1.5, col0 = "white", col1 = "black", newplot = F)
```




```
separationplot(predict(modelAlllogit, dfTest, type = "prob")$war,
               as.numeric(dfTest$warstds)-1, type = "line",
               line = T, lwd2 = 1,
               show.expected = T,
               heading = "All Variables Logit",
               height = 1.5, col0 = "white", col1 = "black", newplot = F)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```




```
separationplot(predict(modelAllrf, dfTest, type = "prob")$war,
               as.numeric(dfTest$warstds)-1, type = "line",
               line = T, lwd2 = 1,
               show.expected = T,
               heading = "All Variables Random Forest",
```

```
          height = 1.5, col0 = "white", col1 = "black", newplot = F)
```

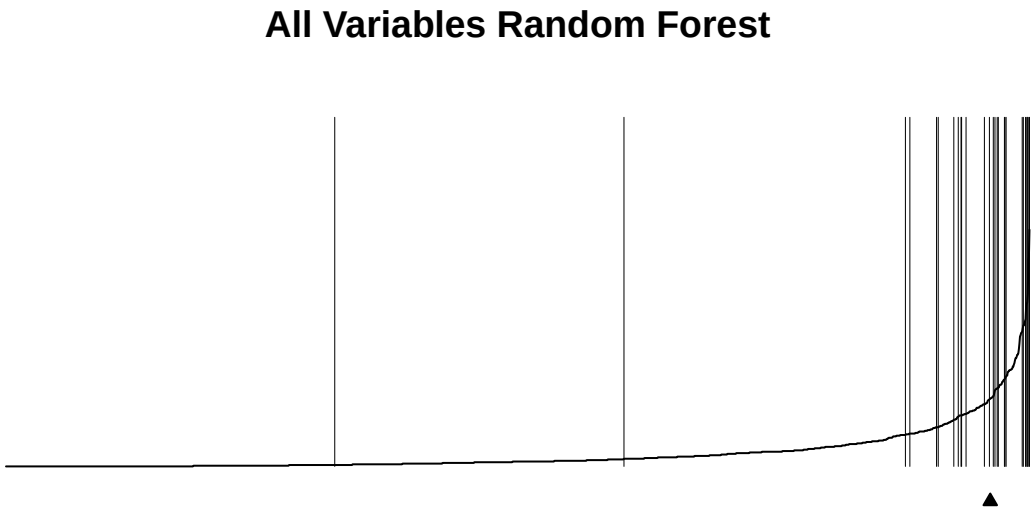


Figure 2: ROC curves for the four models

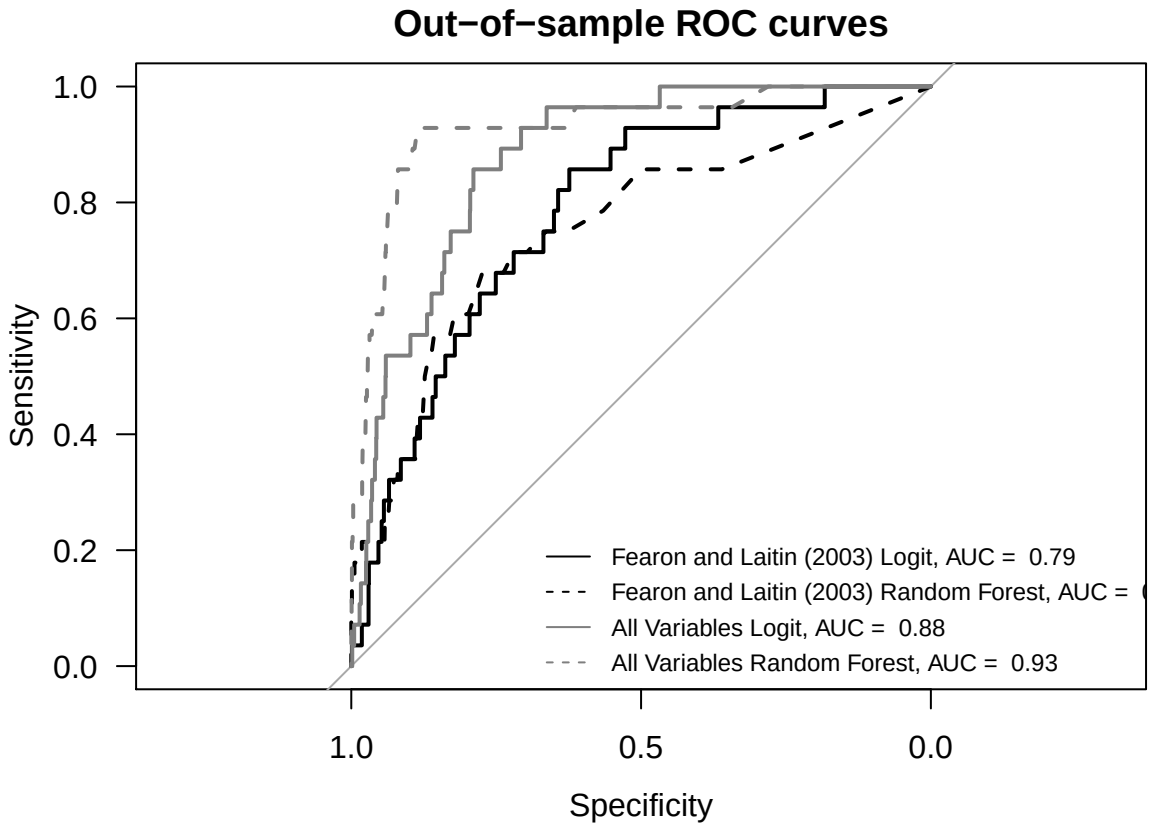
```
plot.roc(dfTest$warstds,
         predict(modelFLlogit, dfTest, type = "prob")$war,
         col = "black", las = 1, xlim = c(1,0), bty = "n",
         main = "Out-of-sample ROC curves")

plot.roc(dfTest$warstds,
         predict(modelFLrf, dfTest, type = "prob")$war,
         add = T,
         col = "black", lty = "dashed")


plot.roc(dfTest$warstds,
         predict(modelAlllogit, dfTest, type = "prob")$war,
         add = T,
         col = "grey 50", lty = "solid")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
plot.roc(dfTest$warstds,
         predict(modelAllrf, dfTest, type = "prob")$war,
         add = T,
         col = "grey 50", lty = "dashed")
legend(0.7, 0.25,
       c(paste("Fearon and Laitin (2003) Logit, AUC = ",
               round(as.numeric(roc(dfTest$warstds,
                                    predict(modelFLlogit, dfTest, type = "prob")$war)$auc),2)),
         paste("Fearon and Laitin (2003) Random Forest, AUC = ",
               round(as.numeric(roc(dfTest$warstds,
                                    predict(modelFLrf, dfTest, type = "prob")$war)$auc),2)),
         paste("All Variables Logit, AUC = ",
               round(as.numeric(roc(dfTest$warstds,
```

```
                                    predict(modelAlllogit, dfTest, type = "prob")$war)$auc),2)),
        paste("All Variables Random Forest, AUC = ",
              round(as.numeric(roc(dfTest$warstds,
                                    predict(modelAllrf, dfTest, type = "prob")$war)$auc),2))),
       lty=c("solid", "dashed", "solid", "dashed"),
       col = c("black","black", "grey 50", "grey 50"), bty="n",
       cex = .75)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```




```
rocFLlogit <- roc(dfTest$warstds,predict(modelFLlogit, dfTest, type = "prob")$war)
rocFLrf <- roc(dfTest$warstds,predict(modelFLrf, dfTest, type = "prob")$war)

rocAlllogit <- roc(dfTest$warstds,predict(modelAlllogit, dfTest, type = "prob")$war)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
rocAllrf <- roc(dfTest$warstds,predict(modelAllrf, dfTest, type = "prob")$war)

set.seed(221)
roc.test(rocAlllogit, rocAllrf, method = "bootstrap")

set.seed(221)
aucCIFLlogit <- ci.auc(rocFLlogit, method = "bootstrap")
set.seed(221)
aucCIFLrf <- ci.auc(rocFLrf, method = "bootstrap")
```

```r
set.seed(221)
aucCIAlllogit <- ci.auc(rocAlllogit, method = "bootstrap")
set.seed(221)
aucCIAllrf <- ci.auc(rocAllrf, method = "bootstrap")
```

Code for Appendix

The four models with up-sampling within the cross-validation procedure

```r
tcup <- trainControl(method = "cv",
                     number = folds,
                     summaryFunction = twoClassSummary,
                     classProb = TRUE,
                     savePredictions = TRUE,
                     seeds = seeds,
                     index = cvIndex,
                     sampling = "up"
)

# The Fearon/Laitin models.

# Logit
modelFLlogitup <- train(warstds ~ warhist + ln_gdpen + lpopns + lmtnest + ncontig + oil
                        + nwstate + inst3 + pol4 + ef + relfrac,
                        metric = "ROC", method = "glm", family = "binomial",
                        trControl = tcup, data = dfTrain
)

summary(modelFLlogitup)

modelFLlogitup

confusionMatrix(modelFLlogitup, norm = "average")


# Random Forest
modelFLrfup <- train(warstds ~ warhist + ln_gdpen + lpopns + lmtnest + ncontig + oil
                     + nwstate + inst3 + pol4 + ef + relfrac,
                     metric = "ROC", method = "rf",
                     trControl = tcup, data = dfTrain,
                     keep.inbag = TRUE
)

modelFLrfup

confusionMatrix(modelFLrfup, norm = "average")



############
```

```
# Logit
modelAlllogitup <- train(warstds ~ .,
                         metric = "ROC", method = "glm", family = "binomial",
                         trControl = tcup, data = dfTrain
)
```

```
## Warning: glm.fit: algorithm did not converge
```

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

```
summary(modelAlllogitup)

modelAlllogitup

confusionMatrix(modelAlllogitup, norm = "average")


# Random Forest
modelAllrfup <- train(warstds ~ .,
                      metric = "ROC", method = "rf",
                      trControl = tcup, data = dfTrain,
                      keep.inbag = TRUE
)

modelAllrfup

confusionMatrix(modelAllrfup, norm = "average")
```

Confusion Matrices

```
table(pred = predict(modelFLlogitup, dfTest, type = "raw"), obs = dfTest$warstds)
```

```
##        obs
## pred    peace  war
##   peace  1332   10
##   war     481   18
```

```
table(pred = predict(modelFLrfup, dfTest, type = "raw"), obs = dfTest$warstds)
```

```
##        obs
## pred    peace  war
##   peace  1744   24
##   war      69    4
```

```
table(pred = predict(modelAlllogitup, dfTest, type = "raw"), obs = dfTest$warstds)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

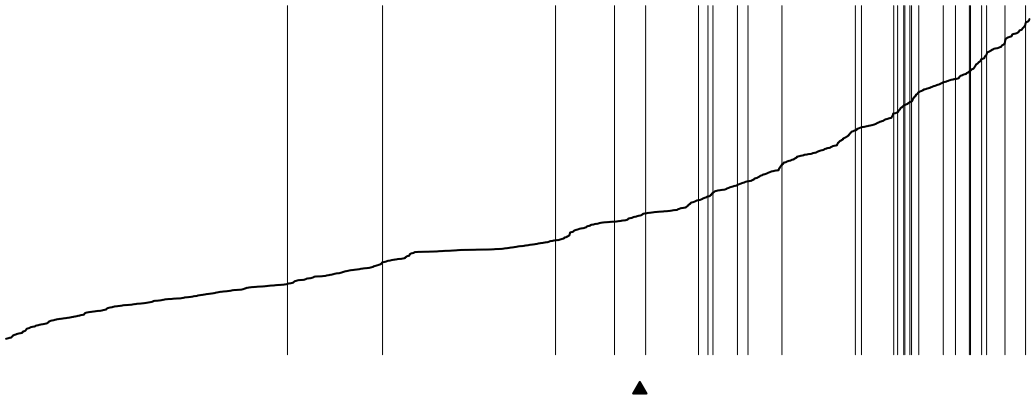
```
##        obs
## pred    peace  war
##   peace  1497    8
##   war     316   20
```

```
table(pred = predict(modelAllrfup, dfTest, type = "raw"), obs = dfTest$warstds)
```

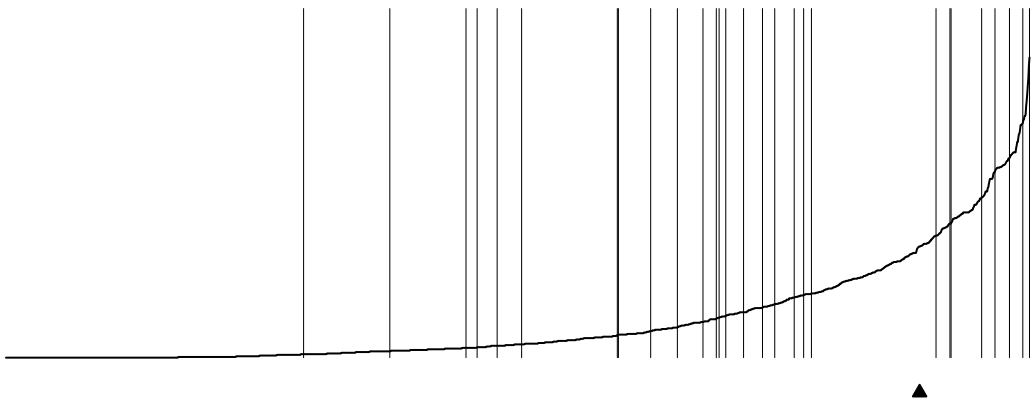
```
##         obs
## pred     peace  war
##   peace  1807   25
##   war       6    3
```

Separation Plots and ROC curves for the models with up-sampling

```
separationplot(predict(modelFLlogitup, dfTest, type = "prob")$war,
               as.numeric(dfTest$warstds)-1, type = "line",
               line = T, lwd2=1,
               show.expected=T,
               heading="Fearon and Laitin (2003) Logit with up-sampling",
               height=1.5, col0="white", col1="black",newplot = F)
```

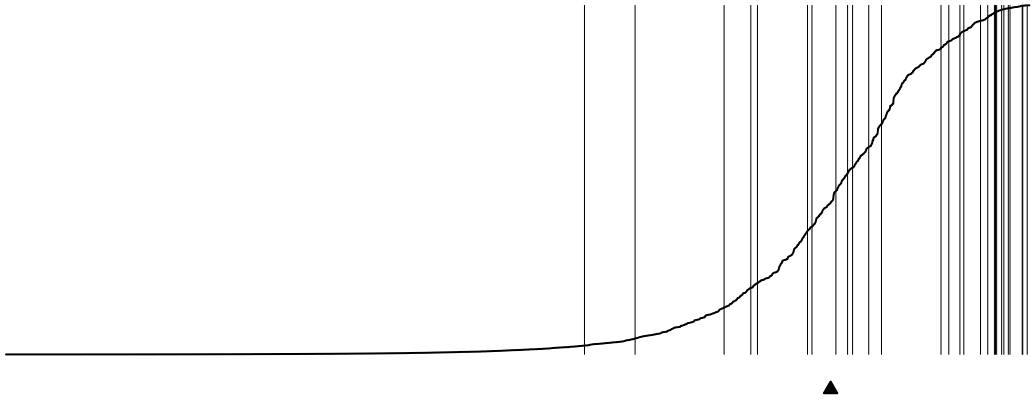



```
separationplot(predict(modelFLrfup, dfTest, type = "prob")$war,
               as.numeric(dfTest$warstds)-1, type = "line",
               line = T, lwd2=1,
               show.expected=T,
               heading="Fearon and Laitin (2003) Random Forest with up-sampling",
               height=1.5, col0="white", col1="black",newplot = F)
```

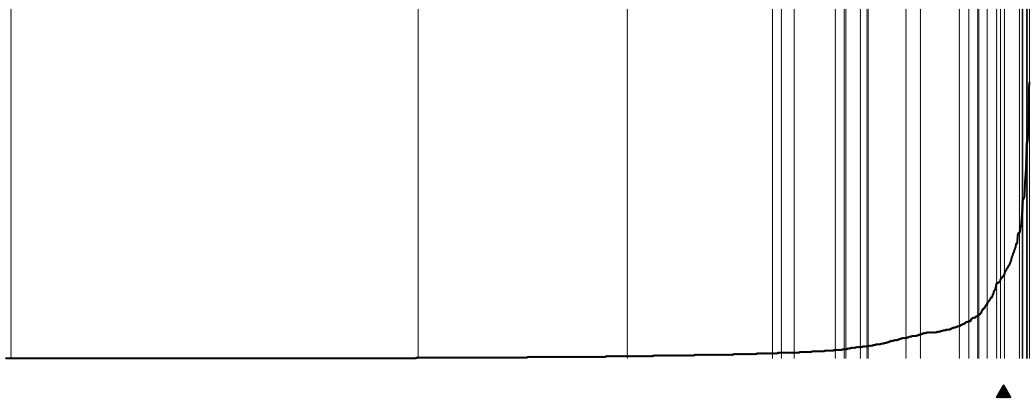


```
separationplot(predict(modelAlllogitup, dfTest, type = "prob")$war,
               as.numeric(dfTest$warstds)-1, type = "line",
               line = T, lwd2=1,
               show.expected=T,
               heading="All Variables Logit with up-sampling",
               height=1.5, col0="white", col1="black",newplot = F)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```




```
separationplot(predict(modelAllrfup, dfTest, type = "prob")$war,
               as.numeric(dfTest$warstds)-1, type = "line",
               line = T, lwd2=1,
               show.expected=T,
               heading="All Variables Random Forest with up-sampling",
               height=1.5, col0="white", col1="black",newplot = F)
```



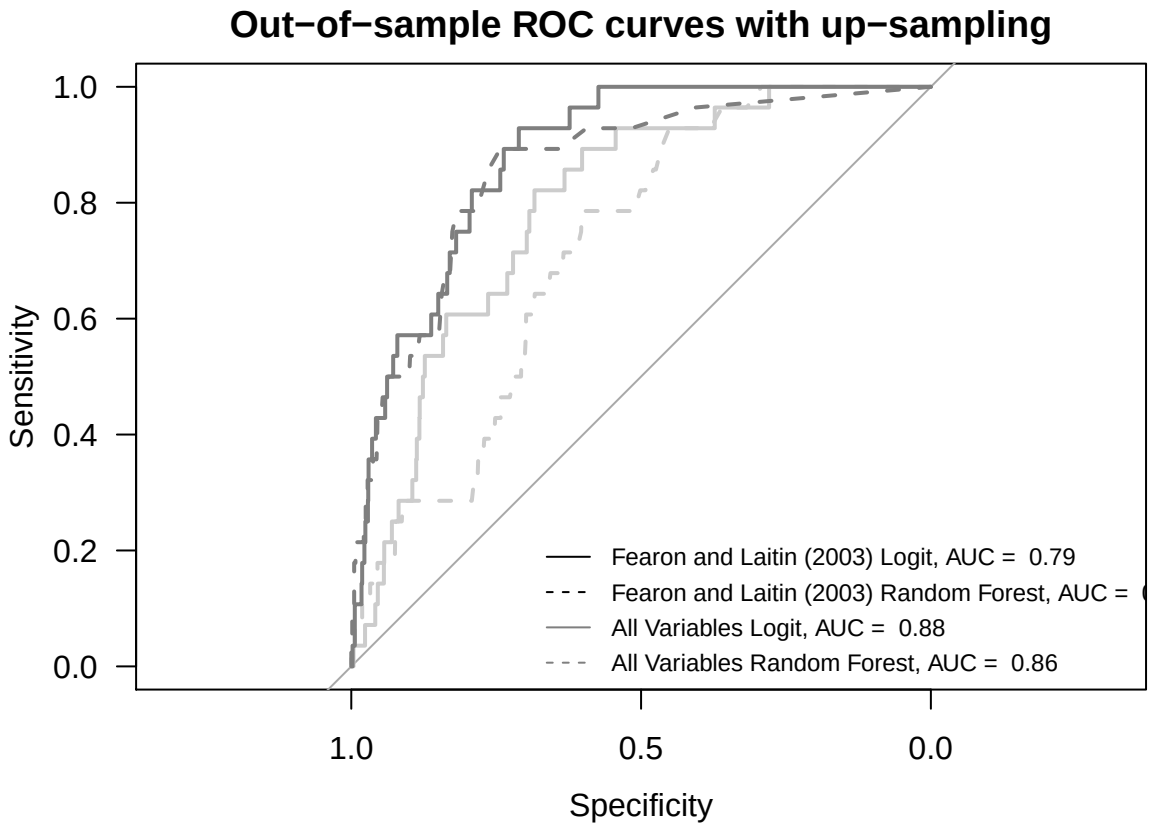

```
plot.roc(dfTest$warstds,
         predict(modelFLlogitup, dfTest, type = "prob")$war,
         col = "grey 80", las = 1, xlim = c(1,0), bty = "n",
         main = "Out-of-sample ROC curves with up-sampling")
```

```
plot.roc(dfTest$warstds,
         predict(modelFLrfup, dfTest, type = "prob")$war,
         add = T,
         col = "grey 80", lty = "dashed")


plot.roc(dfTest$warstds,
         predict(modelAlllogitup, dfTest, type = "prob")$war,
         add = T,
         col = "grey 50", lty = "solid")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
plot.roc(dfTest$warstds,
         predict(modelAllrfup, dfTest, type = "prob")$war,
         add = T,
         col = "grey 50", lty = "dashed")
legend(0.7, 0.25,
       c(paste("Fearon and Laitin (2003) Logit, AUC = ",
              round(as.numeric(roc(dfTest$warstds,
                                   predict(modelFLlogitup, dfTest, type = "prob")$war)$auc),2)),
         paste("Fearon and Laitin (2003) Random Forest, AUC = ",
              round(as.numeric(roc(dfTest$warstds,
                                   predict(modelFLrfup, dfTest, type = "prob")$war)$auc),2)),
         paste("All Variables Logit, AUC = ",
              round(as.numeric(roc(dfTest$warstds,
                                   predict(modelAlllogitup, dfTest, type = "prob")$war)$auc),2)),
         paste("All Variables Random Forest, AUC = ",
              round(as.numeric(roc(dfTest$warstds,
                                   predict(modelAllrfup, dfTest, type = "prob")$war)$auc),2))),
       lty=c("solid", "dashed", "solid", "dashed"),
       col = c("black","black", "grey 50", "grey 50"), bty="n",
       cex = .75)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```



The four models with down-sampling within the cross-validation procedure

```
tcdown <- trainControl(method = "cv",
                       number = folds,
                       summaryFunction = twoClassSummary,
                       classProb = TRUE,
                       savePredictions = TRUE,
                       seeds = seeds,
                       index = cvIndex,
                       sampling = "down"
)

# The Fearon/Laitin models.

# Logit
modelFLlogitdown <- train(warstds ~ warhist + ln_gdpen + lpopns + lmtnest + ncontig + oil
                            + nwstate + inst3 + pol4 + ef + relfrac,
                          metric = "ROC", method = "glm", family = "binomial",
                          trControl = tcdown, data = dfTrain
)

summary(modelFLlogitdown)

modelFLlogitdown

confusionMatrix(modelFLlogitdown, norm = "average")
```

```
# Random Forest
modelFLrfdown <- train(warstds ~ warhist + ln_gdpen + lpopns + lmtnest + ncontig + oil
                       + nwstate + inst3 + pol4 + ef + relfrac,
                     metric = "ROC", method = "rf",
                     trControl = tcdown, data = dfTrain,
                     keep.inbag = TRUE
)

modelFLrfdown

confusionMatrix(modelFLrfdown, norm = "average")



#############

# Logit
modelAlllogitdown <- train(warstds ~ .,
                         metric = "ROC", method = "glm", family = "binomial",
                         trControl = tcdown, data = dfTrain
)
```

```
## Warning: glm.fit: algorithm did not converge
```

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

```
summary(modelAlllogitdown)

modelAlllogitdown

confusionMatrix(modelAlllogitdown, norm = "average")


# Random Forest
modelAllrfdown <- train(warstds ~ .,
                      metric = "ROC", method = "rf",
                      trControl = tcdown, data = dfTrain,
                      keep.inbag = TRUE
)

modelAllrfdown

confusionMatrix(modelAllrfdown, norm = "average")
```

Confusion Matrices

```
table(pred = predict(modelFLlogitdown, dfTest, type = "raw"), obs = dfTest$warstds)
```

```
##        obs
## pred    peace  war
##   peace  1274    9
##   war     539   19
```

```
table(pred = predict(modelFLrfdown, dfTest, type = "raw"), obs = dfTest$warstds)
```

```
##         obs
## pred     peace  war
##   peace  1303   11
##   war     510   17
```

```
table(pred = predict(modelAlllogitdown, dfTest, type = "raw"), obs = dfTest$warstds)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

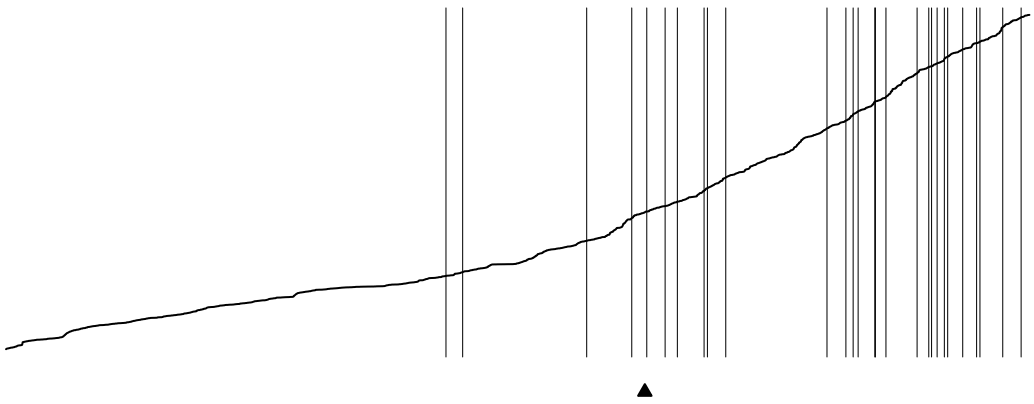
```
##         obs
## pred     peace  war
##   peace  1360    8
##   war     453   20
```

```
table(pred = predict(modelAllrfdown, dfTest, type = "raw"), obs = dfTest$warstds)
```

```
##         obs
## pred     peace  war
##   peace  1579    6
##   war     234   22
```

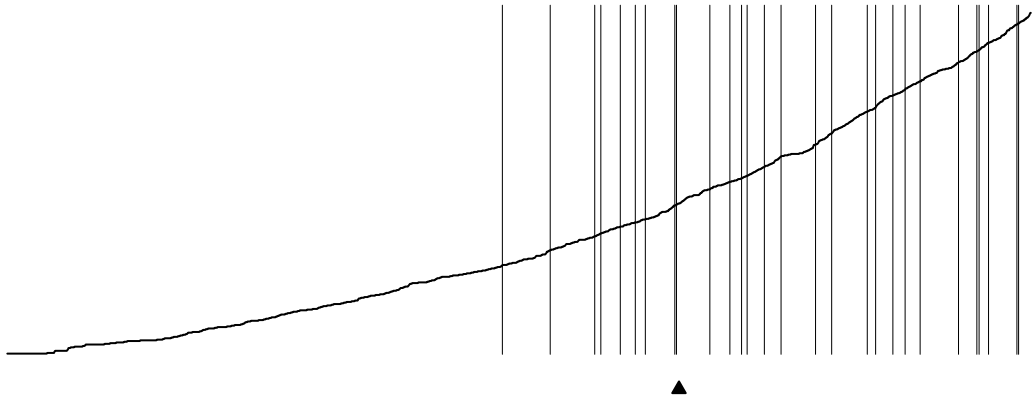
Separation Plots and ROC curves for the models with down-sampling

```
separationplot(predict(modelFLlogitdown, dfTest, type = "prob")$war,
               as.numeric(dfTest$warstds)-1, type = "line",
               line = T, lwd2=1,
               show.expected=T,
               heading="Fearon and Laitin (2003) Logit with down-sampling",
               height=1.5, col0="white", col1="black",newplot = F)
```



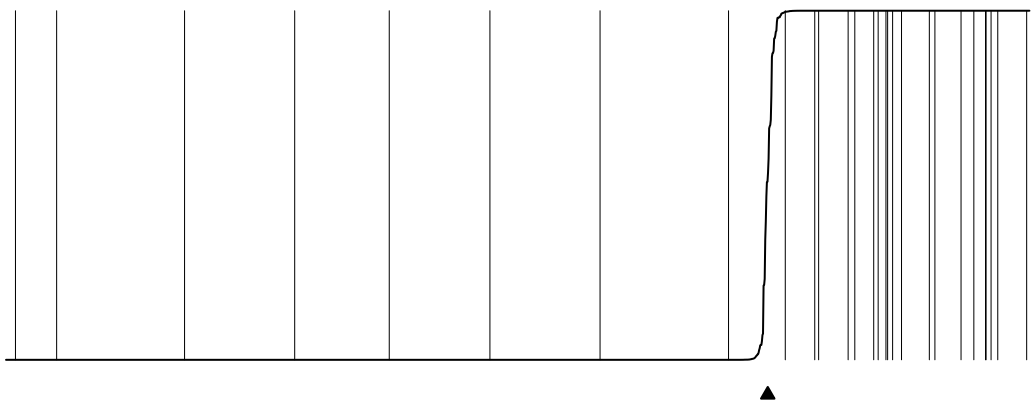

```
separationplot(predict(modelFLrfdown, dfTest, type = "prob")$war,
               as.numeric(dfTest$warstds)-1, type = "line",
               line = T, lwd2=1,
               show.expected=T,
```

```
              heading="Fearon and Laitin (2003) Random Forest with down-sampling",
              height=1.5, col0="white", col1="black",newplot = F)
```

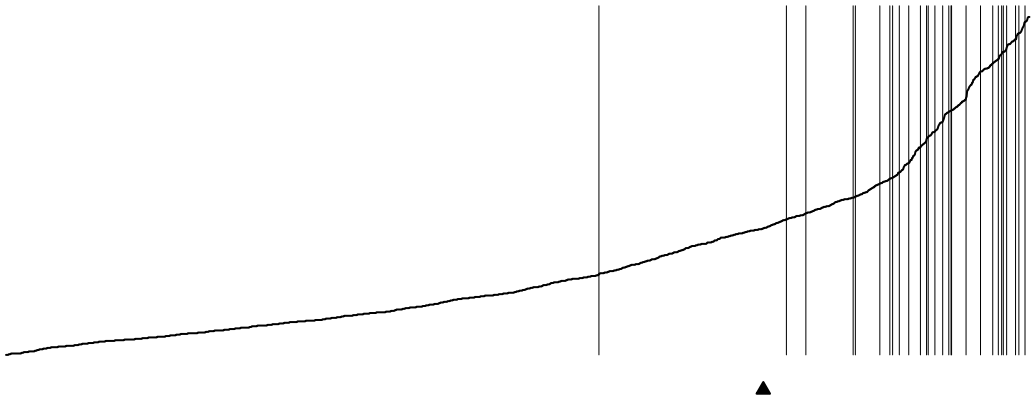



```
separationplot(predict(modelAlllogitdown, dfTest, type = "prob")$war,
               as.numeric(dfTest$warstds)-1, type = "line",
               line = T, lwd2=1,
               show.expected=T,
               heading="All Variables Logit with down-sampling",
               height=1.5, col0="white", col1="black",newplot = F)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```




```
separationplot(predict(modelAllrfdown, dfTest, type = "prob")$war,
               as.numeric(dfTest$warstds)-1, type = "line",
               line = T, lwd2=1,
               show.expected=T,
               heading="All Variables Random Forest with down-sampling",
               height=1.5, col0="white", col1="black",newplot = F)
```

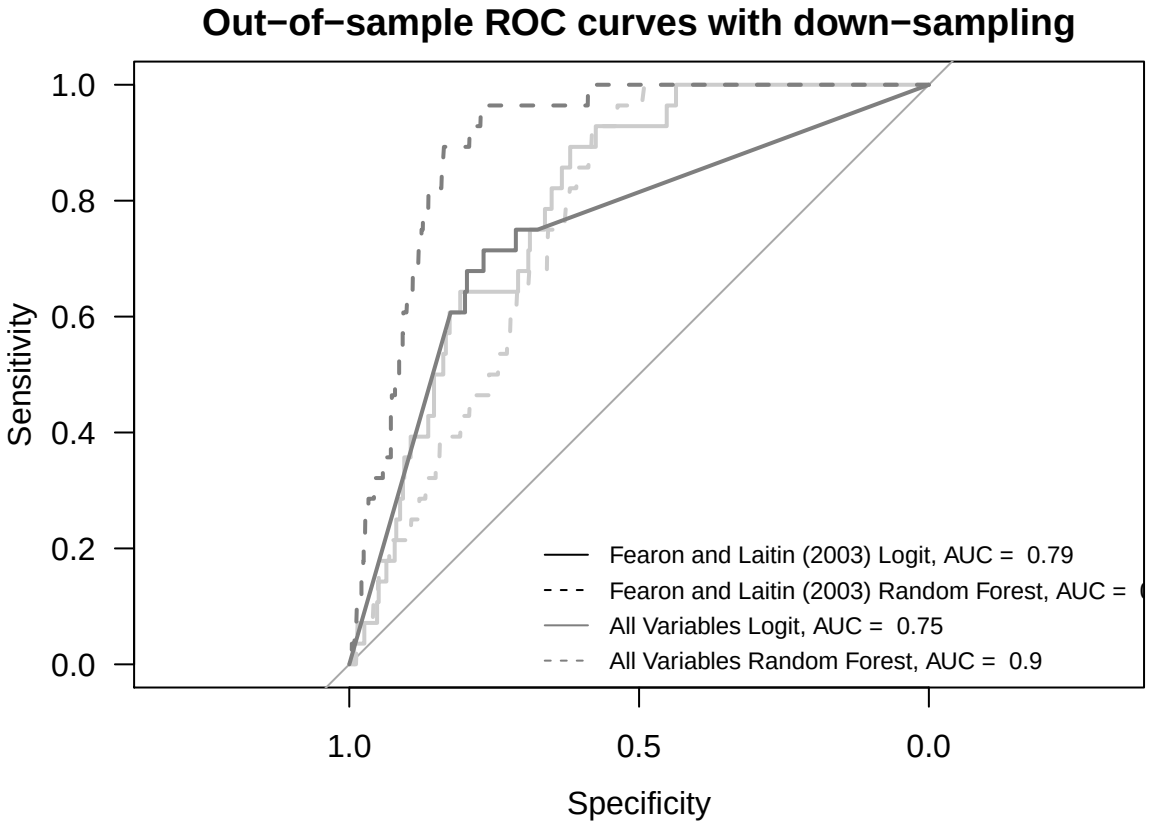



```
plot.roc(dfTest$warstds,
         predict(modelFLlogitdown, dfTest, type = "prob")$war,
         col = "grey 80", las = 1, xlim = c(1,0), bty = "n", main = "Out-of-sample ROC curves with down

plot.roc(dfTest$warstds,
         predict(modelFLrfdown, dfTest, type = "prob")$war,
         add = T,
         col = "grey 80", lty = "dashed")


plot.roc(dfTest$warstds,
         predict(modelAlllogitdown, dfTest, type = "prob")$war,
         add = T,
         col = "grey 50", lty = "solid")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
plot.roc(dfTest$warstds,
         predict(modelAllrfdown, dfTest, type = "prob")$war,
         add = T,
         col = "grey 50", lty = "dashed")
legend(0.7, 0.25,
       c(paste("Fearon and Laitin (2003) Logit, AUC = ",
               round(as.numeric(roc(dfTest$warstds,
                                    predict(modelFLlogitdown, dfTest, type = "prob")$war)$auc),2)),
         paste("Fearon and Laitin (2003) Random Forest, AUC = ",
               round(as.numeric(roc(dfTest$warstds,
                                    predict(modelFLrfdown, dfTest, type = "prob")$war)$auc),2)),
         paste("All Variables Logit, AUC = ",
               round(as.numeric(roc(dfTest$warstds,
                                    predict(modelAlllogitdown, dfTest, type = "prob")$war)$auc),2)),
         paste("All Variables Random Forest, AUC = ",
               round(as.numeric(roc(dfTest$warstds,
                                    predict(modelAllrfdown, dfTest, type = "prob")$war)$auc),2))),
       lty=c("solid", "dashed", "solid", "dashed"),
       col = c("black","black", "grey 50", "grey 50"), bty="n",
       cex = .75)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```



Collier/Hoeffler Logit

```
modelCHlogit <- train(warstds ~ sxpnew+sxpsq+ln_gdpen+gdpgrowth+warhist+lmtnest+ef
                      +popdense+lpopns+coldwar+seceduc+ptime,
                      metric = "ROC", method = "glm", family = "binomial",
                      trControl = tc, data = dfTrain
                      )
```

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

```
summary(modelCHlogit) # Not in paper

modelCHlogit # Not in paper

confusionMatrix(modelCHlogit, norm = "average") # Not in paper
```

Collier/Hoeffler Random Forest

```
modelCHrf <- train(warstds ~ sxpnew+sxpsq+ln_gdpen+gdpgrowth+warhist+lmtnest+ef+popdense
                   +lpopns+coldwar+seceduc+ptime,
                    metric = "ROC", method = "rf",
                    trControl = tc, data = dfTrain,
```

```
                  keep.inbag = TRUE
                  )

modelCHrf # Not in paper

confusionMatrix(modelCHrf, norm = "average") # Not in paper
```

The plots and confusion matrices to compare the models. Collier/Hoeffler vs. all 90 variables

Confusion Matrices for the models at a threshold for positive prediction of 0.5

```
table(pred = predict(modelCHlogit, dfTest, type = "raw"), obs = dfTest$warstds)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
##        obs
## pred    peace  war
##   peace  1813   28
##   war       0    0
```

```
table(pred = predict(modelCHrf, dfTest, type = "raw"), obs = dfTest$warstds)
```

```
##        obs
## pred    peace  war
##   peace  1807   19
##   war       6    9
```

```
table(pred = predict(modelAlllogit, dfTest, type = "raw"), obs = dfTest$warstds)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
##        obs
## pred    peace  war
##   peace  1788   26
##   war      25    2
```

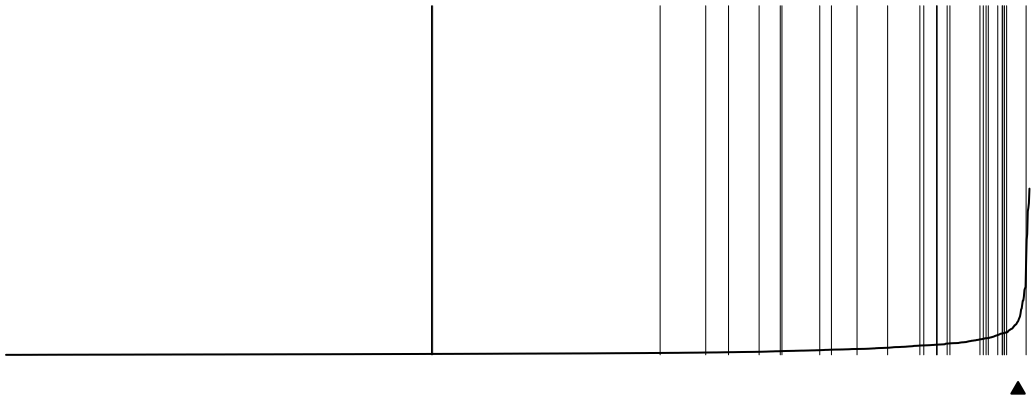
```
table(pred = predict(modelAllrf, dfTest, type = "raw"), obs = dfTest$warstds)
```

```
##        obs
## pred    peace  war
##   peace  1812   26
##   war       1    2
```

Separation Plots for the four models (Collier/Hoeffler)

```
separationplot(predict(modelCHlogit, dfTest, type = "prob")$war,
               as.numeric(dfTest$warstds)-1, type = "line",
               line = T, lwd2 = 1,
               show.expected = T,
               heading = "Collier and Hoeffler Variables (2004) Logit",
               height = 1.5, col0 = "white", col1 = "black", newplot = F)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

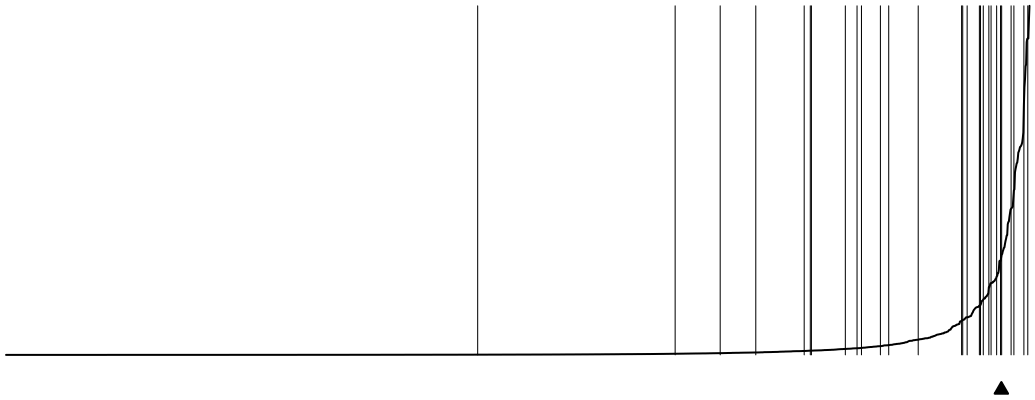



```
separationplot(predict(modelCHrf, dfTest, type = "prob")$war,
               as.numeric(dfTest$warstds)-1, type = "line",
               line = T, lwd2 = 1,
               show.expected = T,
               heading = "Collier and Hoeffler Variables (2004) Random Forest",
               height = 1.5, col0 = "white", col1 = "black", newplot = F)
```


```
separationplot(predict(modelAlllogit, dfTest, type = "prob")$war,
               as.numeric(dfTest$warstds)-1, type = "line",
               line = T, lwd2 = 1,
               show.expected = T,
               heading = "All Variables Logit",
               height = 1.5, col0 = "white", col1 = "black", newplot = F)
```

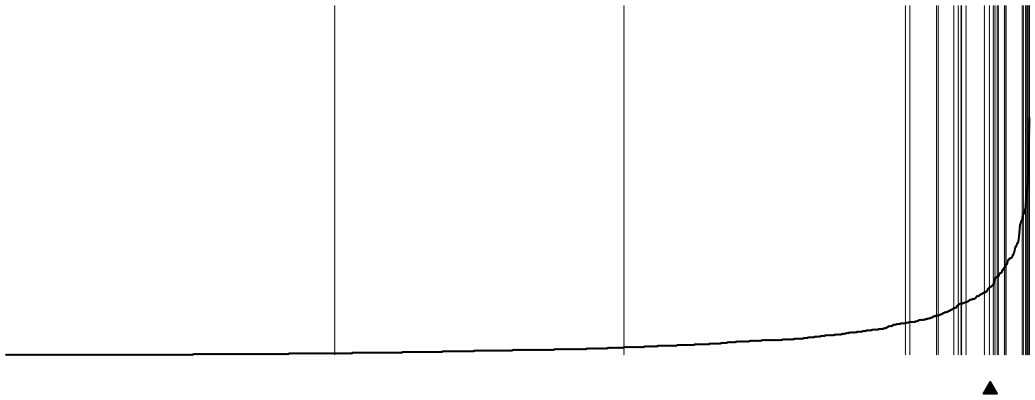
```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

All Variables Logit



```
separationplot(predict(modelAllrf, dfTest, type = "prob")$war,
               as.numeric(dfTest$warstds)-1, type = "line",
               line = T, lwd2 = 1,
               show.expected = T,
               heading = "All Variables Random Forest",
               height = 1.5, col0 = "white", col1 = "black", newplot = F)
```

All Variables Random Forest



ROC curves for the four models (Collier/Hoeffler)

```
plot.roc(dfTest$warstds,
         predict(modelCHlogit, dfTest, type = "prob")$war,
         col = "grey 80", las = 1, xlim = c(1,0), bty = "n",
         main = "Out-of-sample ROC curves")
```

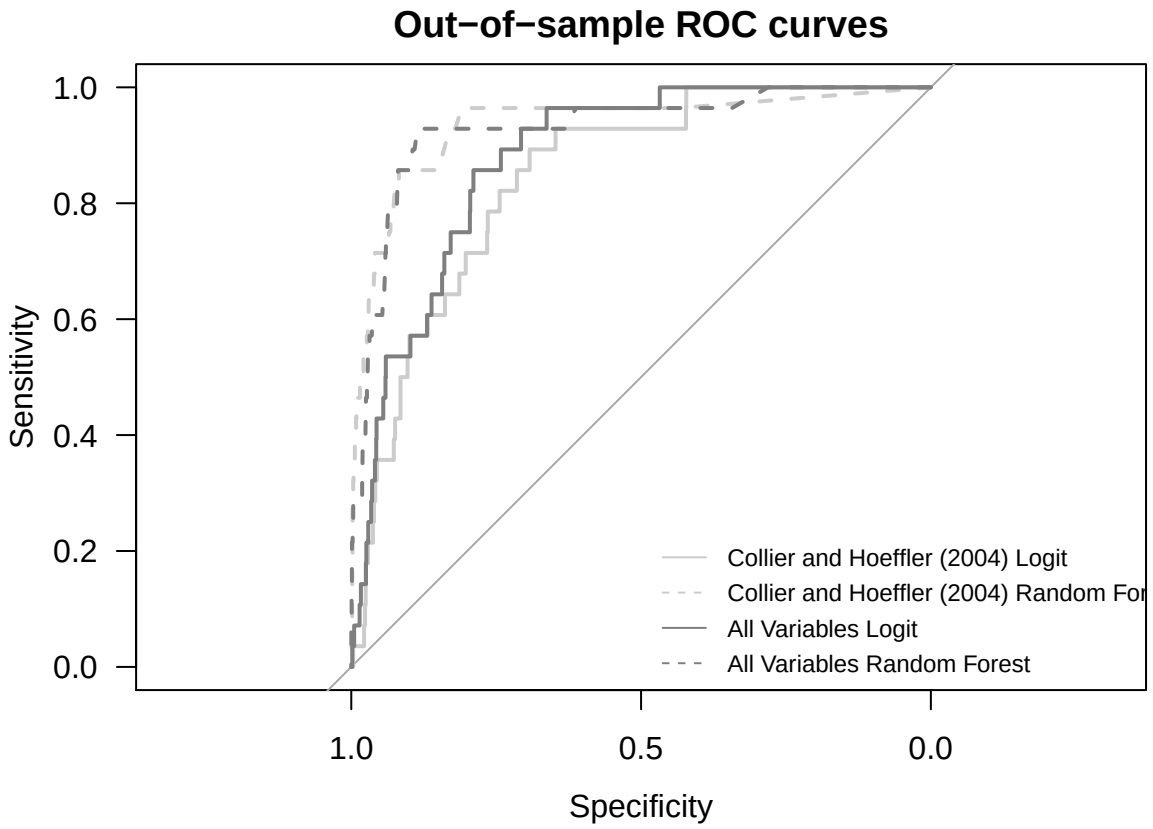
```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
plot.roc(dfTest$warstds,
         predict(modelCHrf, dfTest, type = "prob")$war,
         add = T,
         col = "grey 80", lty = "dashed")
```

```
plot.roc(dfTest$warstds,
         predict(modelAlllogit, dfTest, type = "prob")$war,
         add = T,
         col = "grey 50", lty = "solid")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
plot.roc(dfTest$warstds,
         predict(modelAllrf, dfTest, type = "prob")$war,
         add = T,
         col = "grey 50", lty = "dashed")
legend(0.5, 0.25, c("Collier and Hoeffler (2004) Logit",
                    "Collier and Hoeffler (2004) Random Forest",
                     "All Variables Logit",
                    "All Variables Random Forest" ),
       lty=c("solid", "dashed", "solid", "dashed"),
       col = c("grey 80","grey 80", "grey 50", "grey 50"), bty="n",
       cex = .75)
```



Hegre/Sambanis model specifications

Logit

```
modelHSlogit <- train(warstds ~ lpopns+ln_gdpen+inst3+parreg+geo34+proxregc+gdpgrowth
                      +anoc+partfree+nat_war+lmtnest+decade1+pol4sq+nwstate
                      +regd4_alt+etdo4590+milper+geo1+tnatwar+presi,
                      metric = "ROC", method = "glm", family = "binomial",
                      trControl = tc, data = dfTrain
                      )

summary(modelHSlogit) # Not in paper

modelHSlogit # Not in paper

confusionMatrix(modelHSlogit, norm = "average") # Not in paper
```

Random Forest

```
modelHSrf <- train(warstds ~ lpopns+ln_gdpen+inst3+parreg+geo34+proxregc+gdpgrowth+anoc
                   +partfree+nat_war+lmtnest+decade1+pol4sq+nwstate+regd4_alt+etdo4590
                   +milper+geo1+tnatwar+presi,
                   metric = "ROC", method = "rf",
                   trControl = tc, data = dfTrain,
                   keep.inbag = TRUE
                   )

modelHSrf # Not in paper

confusionMatrix(modelHSrf, norm = "average") # Not in paper
```

The plots and confusion matrices to compare the models. Hegre/Sambanis vs. all 90 variables

Confusion Matrices for the models at a threshold for positive prediction of 0.5

```
table(pred = predict(modelHSlogit, dfTest, type = "raw"), obs = dfTest$warstds)
```

```
##        obs
## pred    peace  war
##   peace  1811   27
##   war       2    1
```

```
table(pred = predict(modelHSrf, dfTest, type = "raw"), obs = dfTest$warstds)
```

```
##        obs
## pred    peace  war
##   peace  1813   27
##   war       0    1
```

```
table(pred = predict(modelAlllogit, dfTest, type = "raw"), obs = dfTest$warstds)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

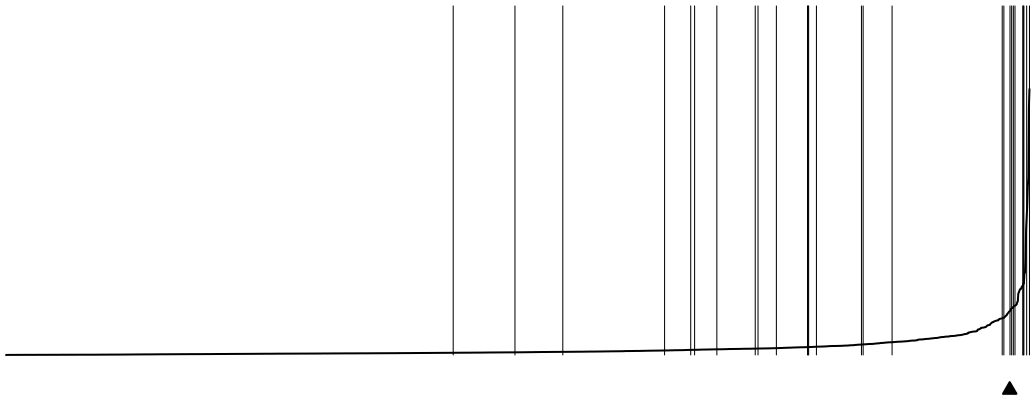
```
##        obs
## pred    peace  war
##   peace  1788   26
##   war      25    2
```

```
table(pred = predict(modelAllrf, dfTest, type = "raw"), obs = dfTest$warstds)
```

```
##        obs
## pred    peace  war
##   peace  1812   26
##   war       1    2
```

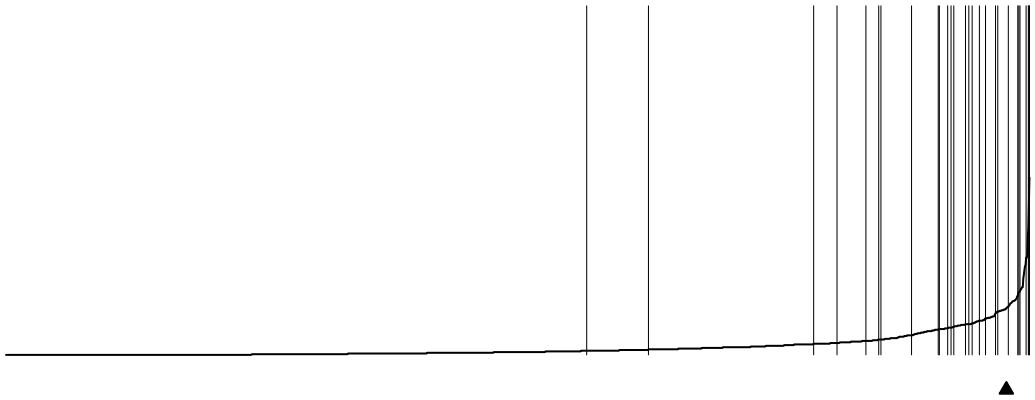
Separation Plots for the four models (Hegre/Sambanis)

```
separationplot(predict(modelHSlogit, dfTest, type = "prob")$war,
               as.numeric(dfTest$warstds)-1, type = "line",
               line = T, lwd2 = 1,
               show.expected = T,
               heading = "Hegre and Sambanis Variables (2006) Logit",
               height = 1.5, col0 = "white", col1 = "black", newplot = F)
```




```
separationplot(predict(modelHSrf, dfTest, type = "prob")$war,
               as.numeric(dfTest$warstds)-1, type = "line",
               line = T, lwd2 = 1,
               show.expected = T,
               heading = "Hegre and Sambanis Variables (2006) Random Forest",
               height = 1.5, col0 = "white", col1 = "black", newplot = F)
```

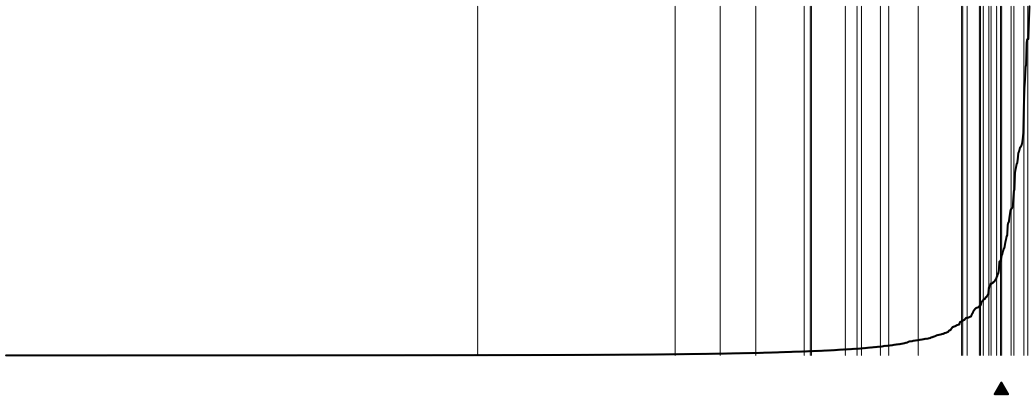
Hegre and Sambanis Variables (2006) Random Forest



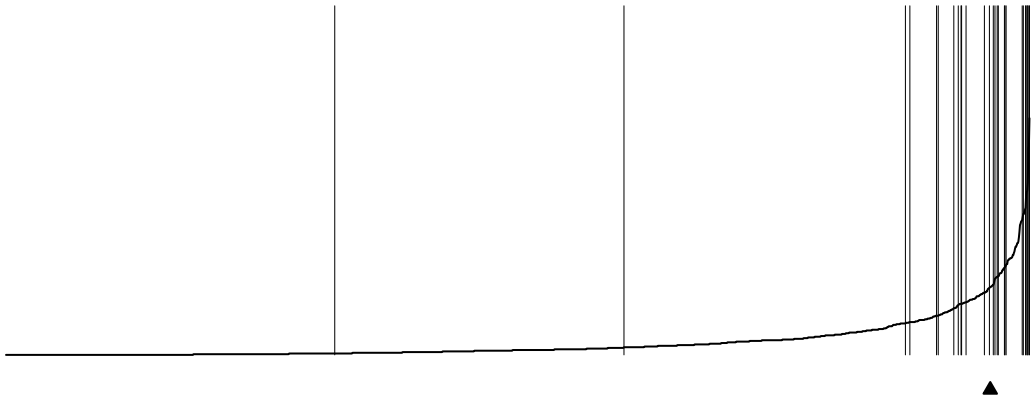
```
separationplot(predict(modelAlllogit, dfTest, type = "prob")$war,
               as.numeric(dfTest$warstds)-1, type = "line",
               line = T, lwd2 = 1,
               show.expected = T,
               heading = "All Variables Logit",
               height = 1.5, col0 = "white", col1 = "black", newplot = F)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

All Variables Logit



```
separationplot(predict(modelAllrf, dfTest, type = "prob")$war,
               as.numeric(dfTest$warstds)-1, type = "line",
               line = T, lwd2 = 1,
               show.expected = T,
               heading = "All Variables Random Forest",
               height = 1.5, col0 = "white", col1 = "black", newplot = F)
```

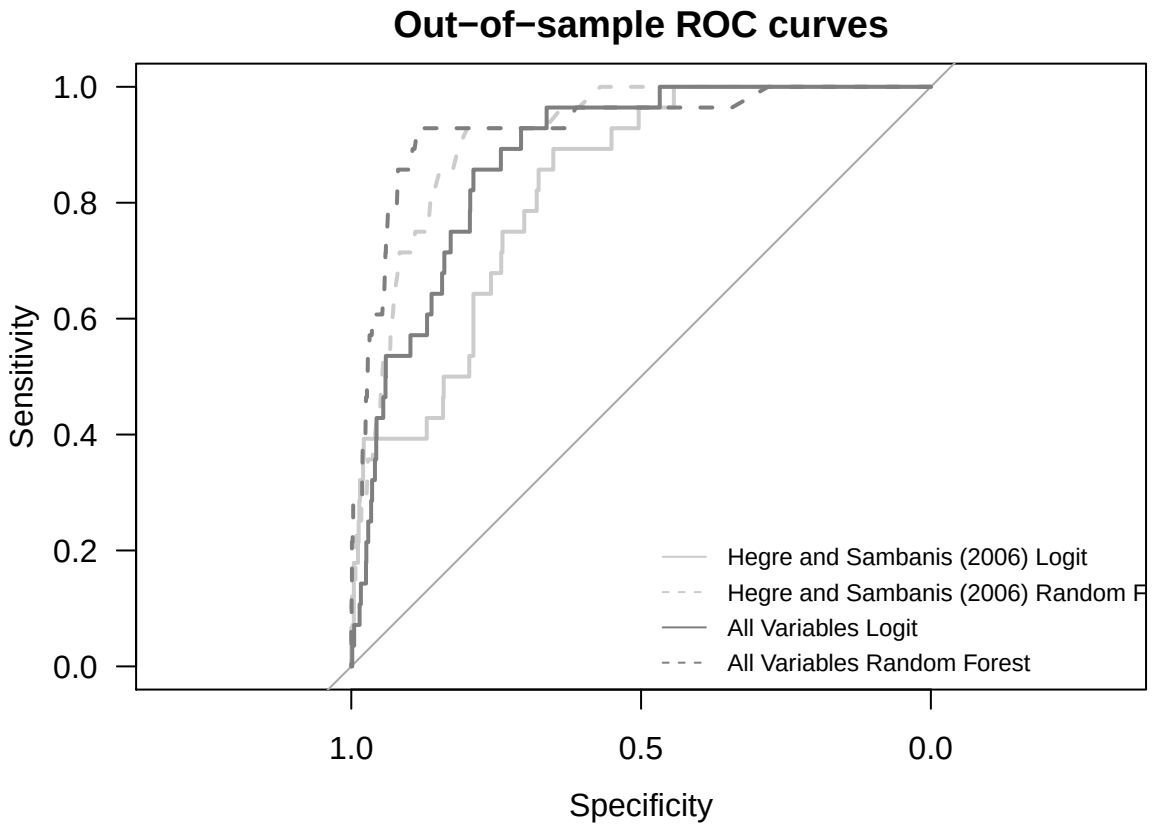


ROC curves for the four models (Hegre/Sambanis)

```
plot.roc(dfTest$warstds,
         predict(modelHSlogit, dfTest, type = "prob")$war,
         col = "grey 80", las = 1, xlim = c(1,0), bty = "n",
         main = "Out-of-sample ROC curves")

plot.roc(dfTest$warstds,
         predict(modelHSrf, dfTest, type = "prob")$war,
         add = T,
         col = "grey 80", lty = "dashed")


plot.roc(dfTest$warstds,
         predict(modelAlllogit, dfTest, type = "prob")$war,
         add = T,
         col = "grey 50", lty = "solid")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
plot.roc(dfTest$warstds,
         predict(modelAllrf, dfTest, type = "prob")$war,
         add = T,
         col = "grey 50", lty = "dashed")
legend(0.5, 0.25, c("Hegre and Sambanis (2006) Logit",
                    "Hegre and Sambanis (2006) Random Forest",
                    "All Variables Logit",
                    "All Variables Random Forest" ),
       lty=c("solid", "dashed", "solid", "dashed"),
       col = c("grey 80","grey 80", "grey 50", "grey 50"), bty="n",
       cex = .75)
```


Out−of−sample ROC curves
Sensitivity
1.0
0.8
0.6
0.4
0.2
0.0
1.0
0.5
0.0
Specificity
Hegre and Sambanis (2006) Logit
Hegre and Sambanis (2006) Random F
All Variables Logit
All Variables Random Forest