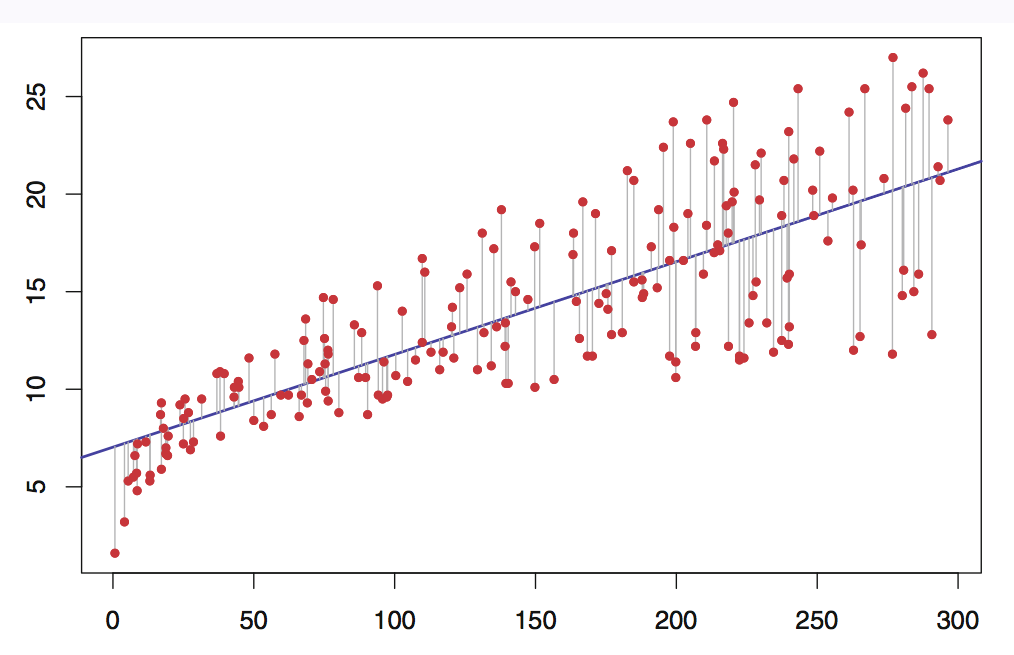
# R Statistics Demo - Model Comparison

The following demo will compare 4 statistical models: 1. Linear Regression 2. Decision Tree 3. Random Forest 4. Neural Network 5. Boosted Regression Tree

# Load historical data

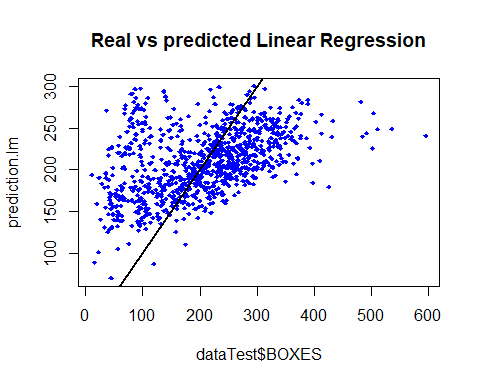
# Set seed and determine training and test data

# Create the Linear regression model



Linear Regression - Least Squares

## [1] 84.03009

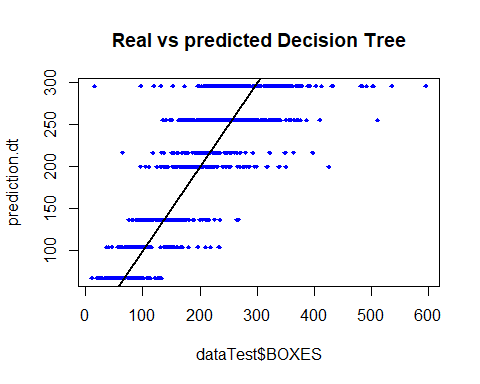


# Create the decision tree model

model.dt <- rpart(data\_formula\_boxes, method = "anova", data=dataTrain)  
prediction.dt <- predict(model.dt, newdata = dataTest)  
rmse.dt <- sqrt(mean((dataTest$BOXES-prediction.dt)^2))  
rmse.dt

## [1] 54.70869

plot(dataTest$BOXES,prediction.dt,col='blue',main='Real vs predicted Decision Tree',pch=18, cex=0.7)  
abline(0,1,lwd=2)

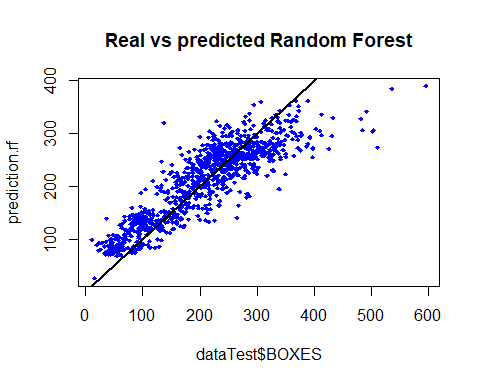


# Create the random forest model

dataTrain\_fac <- dataTrain %>% mutate\_if(is.character, as.factor)  
dataTest\_fac <- dataTest %>% mutate\_if(is.character, as.factor)  
model.rf <- randomForest(data\_formula\_boxes, data = dataTrain\_fac)  
prediction.rf <- predict(model.rf, newdata = dataTest\_fac)  
rmse.rf <- sqrt(mean((dataTest\_fac$BOXES-prediction.rf)^2))  
rmse.rf

## [1] 46.32067

plot(dataTest$BOXES,prediction.rf,col='blue',main='Real vs predicted Random Forest',pch=18, cex=0.7)  
abline(0,1,lwd=2)



# Create the neural net mode

# library('neuralnet')  
#   
# maxs <- apply(dataTrain, 2, max)  
# mins <- apply(dataTrain, 2, min)  
# scaled\_train <- as.data.frame(scale(dataTrain, center = mins, scale = maxs - mins))  
# scaled\_test <- as.data.frame(scale(dataTest, center = mins, scale = maxs - mins))  
# n <- names(scaled\_train)  
# f <- as.formula(paste("BOXES ~", paste(n[!n %in% "BOXES"], collapse = " + ")))  
#   
# nn <- neuralnet(f,data=scaled\_train,hidden=c(8,4),linear.output=F)  
# pr.nn <- compute(nn,scaled\_test[,0:11])  
# pr.nn\_ <- pr.nn$net.result\*(max(data$BOXES)-min(data$BOXES))+min(data$BOXES)  
# test.r <- (scaled\_test$BOXES)\*(max(data$BOXES)-min(data$BOXES))+min(data$BOXES)  
# rmse.nn <- sqrt(mean((dataTest$BOXES-pr.nn\_)^2))  
# rmse.nn  
#   
# plot(dataTest$BOXES,pr.nn\_,col='blue',main='Real vs predicted Neural Net',pch=18, cex=0.7)  
# abline(0,1,lwd=2)

# Create the boosted random forest model

tic()  
model.xgb <- xgb.train(data=xgTrain\_box, objective='reg:linear', eval\_metric='rmse', booster='gbtree', watchlist = list(train=xgTrain\_box, validate=xgVal\_box),   
 early\_stopping\_rounds=100, nrounds = 100000, num\_parallel\_tree=20, print\_every\_n = 20, nthread=8,eta = .1, max\_depth = 7)

## [1] train-rmse:206.662491 validate-rmse:206.324921   
## Multiple eval metrics are present. Will use validate\_rmse for early stopping.  
## Will train until validate\_rmse hasn't improved in 100 rounds.  
##   
## [21] train-rmse:49.004692 validate-rmse:51.825565   
## [41] train-rmse:37.739918 validate-rmse:44.912476   
## [61] train-rmse:34.274696 validate-rmse:44.728710   
## [81] train-rmse:32.041958 validate-rmse:45.088673   
## [101] train-rmse:30.619726 validate-rmse:45.431015   
## [121] train-rmse:28.945339 validate-rmse:45.697258   
## [141] train-rmse:27.396843 validate-rmse:46.126431   
## Stopping. Best iteration:  
## [55] train-rmse:35.154510 validate-rmse:44.669308

toc()

## 40.54 sec elapsed

dataTest.xgb <- build.x(data\_formula\_boxes, data=dataTest, contrasts = FALSE, sparse = TRUE)  
  
prediction.xgb <- predict(model.xgb, newdata = dataTest.xgb)  
plot(dataTest$BOXES,prediction.xgb,col='blue',main='Real vs predicted Boosted Forest',pch=18, cex=0.7)  
 abline(0,1,lwd=2)

