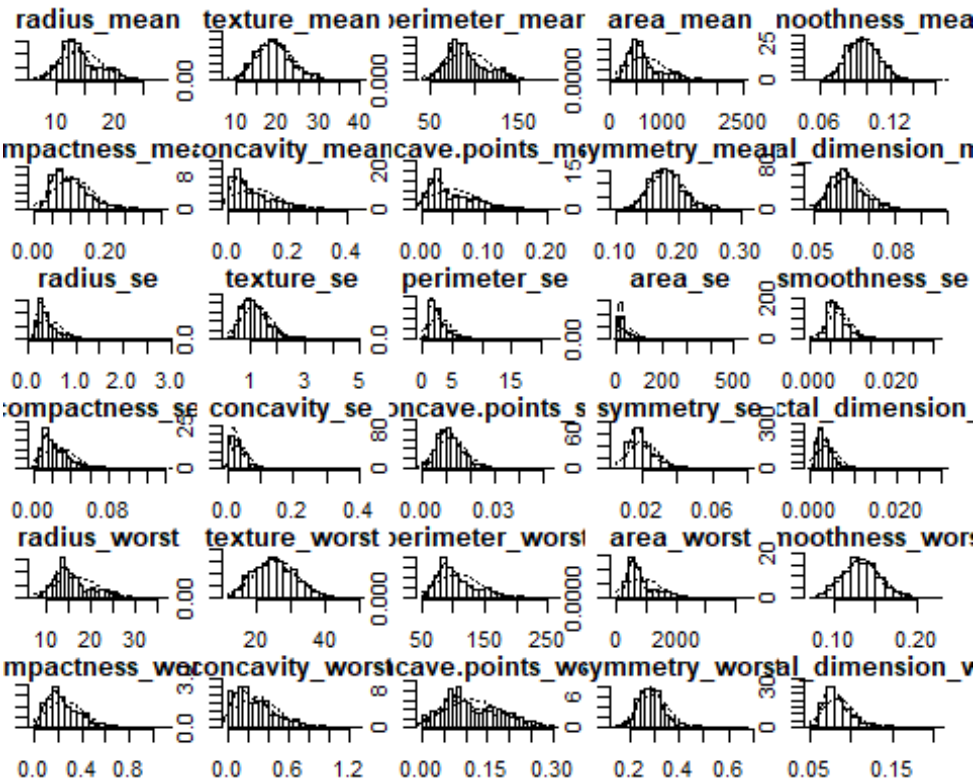
Decision Trees

Brian Seggebruch

March 2, 2019

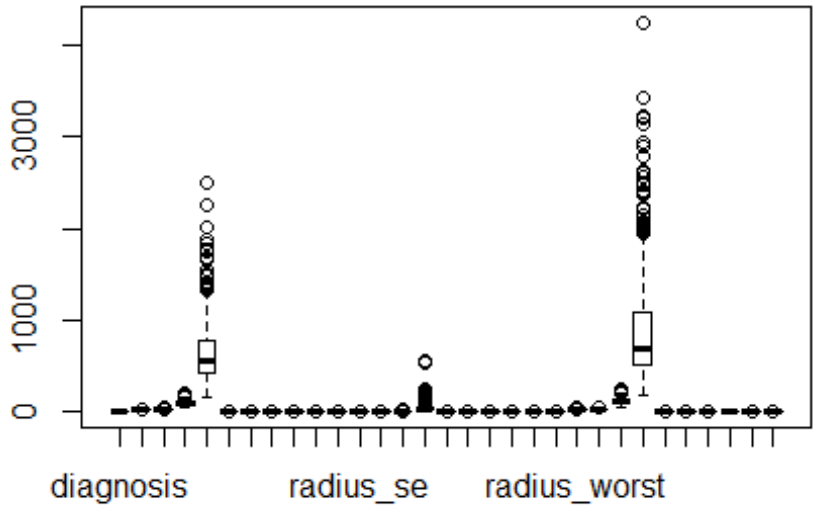
Data prep (output not shown in knitted document for the sake of space):

```
# Data cleaning
breast_cancer<-read.csv("wisconsin_breast_cancer.csv", header = TRUE)
breast_cancer<-breast_cancer[2:32]
breast_cancer_varnames<-read.csv("variable_names.csv", header = TRUE)
breast_cancer
head(breast_cancer)
names(breast_cancer)

is.na(breast_cancer)
breast_cancer[!complete.cases(breast_cancer),]

# Histogram distribution
multi.hist(breast_cancer[,sapply(breast_cancer, is.numeric)])
```

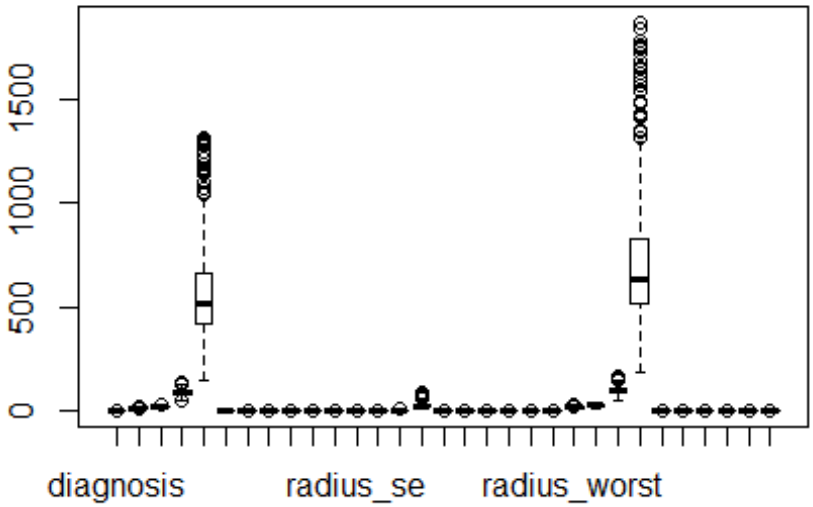



```
# Identify and remove outliers
breast_cancer.cat<-breast_cancer[1]
breast_cancer.num<-breast_cancer[2:31]
```

```r
remove_outliers <- function(x, na.rm = TRUE, ...) {
  qnt <- quantile(x, probs=c(.25, .75), na.rm = na.rm, ...)
  H <- 1.5 * IQR(x, na.rm = na.rm)
  y <- x
  y[x < (qnt[1] - H)] <- NA
  y[x > (qnt[2] + H)] <- NA
  y
}
breast_cancer.noout.ma <- apply(breast_cancer[2:31], 2, remove_outliers)
breast_cancer.noout.num<-data.frame(breast_cancer.noout.ma)
breast_cancer.noout<-cbind(breast_cancer.cat,breast_cancer.noout.num)
breast_cancer.noout.nona<-na.omit(breast_cancer.noout)
boxplot(breast_cancer)
```




```r
boxplot(breast_cancer.noout.nona)
```



Our dataset contains diagnostic data for 579 cancer-screenings, digitized from images of a "fine needle aspirate of mass" procedure. The data is provided by the University of Wisconsin and is intended to be used to help predict whether a mass of cells is malignant or benign. There are ten real-valued variables measured for each record. They are, (from the dataset documentation):

1) radius (mean of distances from center to points on the perimeter)
2) texture (standard deviation of gray-scale values)
3) perimeter
4) area
5) smoothness (local variation in radius lengths)
6) compactness (perimeter^2 / area - 1.0)
7) concavity (severity of concave portions of the contour)
8) concave points (number of concave portions of the contour)
9) symmetry
10) fractal dimension ("coastline approximation" - 1)

Each of these measures is a taken from the cell nuclei present in the image generated from the procedure. The mean, standard error, and "worst" (largest) value are calculated for each image and recorded. Therefore, we have 30 variables (10 real-value measurements * 3 statistically derived values). Each variable is recorded with four significant digits. There are 357 benign classifications and 212 malignant classifications.

We want to understand our data, so we preview it with a few R functions.

```
names(breast_cancer)
head(breast_cancer)
summary(breast_cancer)
```

Splitting into test/train:

```
set.seed(231654)

## 75% of the sample size
sample_size <- floor(0.75 * nrow(breast_cancer))

set.seed(2356498)
train_index <- sample(seq_len(nrow(breast_cancer)), size = sample_size)

train <- breast_cancer[train_index, ]
test <- breast_cancer[-train_index, ]
```

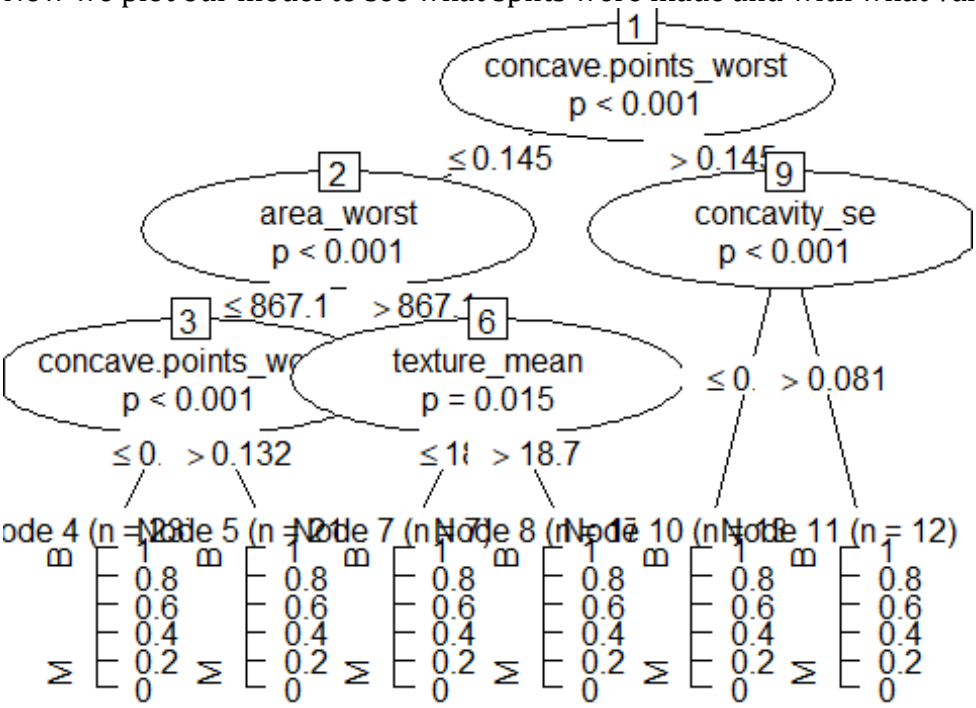
Here we do some minor data cleansing, removing fields we don't want in our model:

```
train$id <- NULL
train$X <- NULL
test$id <- NULL
test$X <- NULL
```

Fitting our model. The Gini Impurity algorithm helps us choose splits that build our tree to be the most accuracte and most direct. The actual formual can be written as: $\sum_{i=1}^{J} {p_i}^2$, where p_i is the fraction of items labeled with class i and J is the number of classes we can choose from:

```
set.seed(3212)
fit <- ctree(diagnosis ~ ., data = train)
```

Now we plot our model to see what splits were made and with what values:



Looking at one specific path from the tree that was created, we can follow the logic and see that our output given these filtering parameters is more favorble for predicting a classification:

```
train[train$concave.points_worst<=0.142 & train$area_worst>947.9,1]

##  [1] B M M B M M M M M M M M M M M M M
## Levels: B M
```

Measuring accuracy:

```
testoutput <- as.matrix(as.character(predict(fit, newdata = test)))
(model_accuracy <- mean(testoutput == test$diagnosis))

## [1] 0.951049
```

Showing the side-by-side of our predicted vs. the actual classifications:

```
sidebyside <- as.data.frame(as.character(test$diagnosis))
sidebyside$predicted <- testoutput
names(sidebyside) <- c('observed', 'predicted')
sidebyside
```

Testing the hypothesis: Our explanatory variables have a significant impact on the outcome of classification…:

```r
probSuccesss <- summary(test$diagnosis)[1]/sum(summary(test$diagnosis))

randomClass_B <- rbinom(10000, 89, probSuccesss)
randomClass_M <- rbinom(10000, 143-89, 1-probSuccesss)
randomClass <- randomClass_B + randomClass_M
randomClass <- as.data.frame(randomClass)
randomClass$accuracy <- randomClass$randomClass/143

length(filter(randomClass, accuracy >= model_accuracy)[,1])
```

```
## [1] 0
```

```r
mu <-  mean(randomClass$randomClass)
stdev <- sd(randomClass$randomClass)
qnorm(0.95, mean = mu, sd = stdev)
```

```
## [1] 87.00274
```