

# Isolation trees

ANOMALY DETECTION IN R



**Alastair Rushworth**

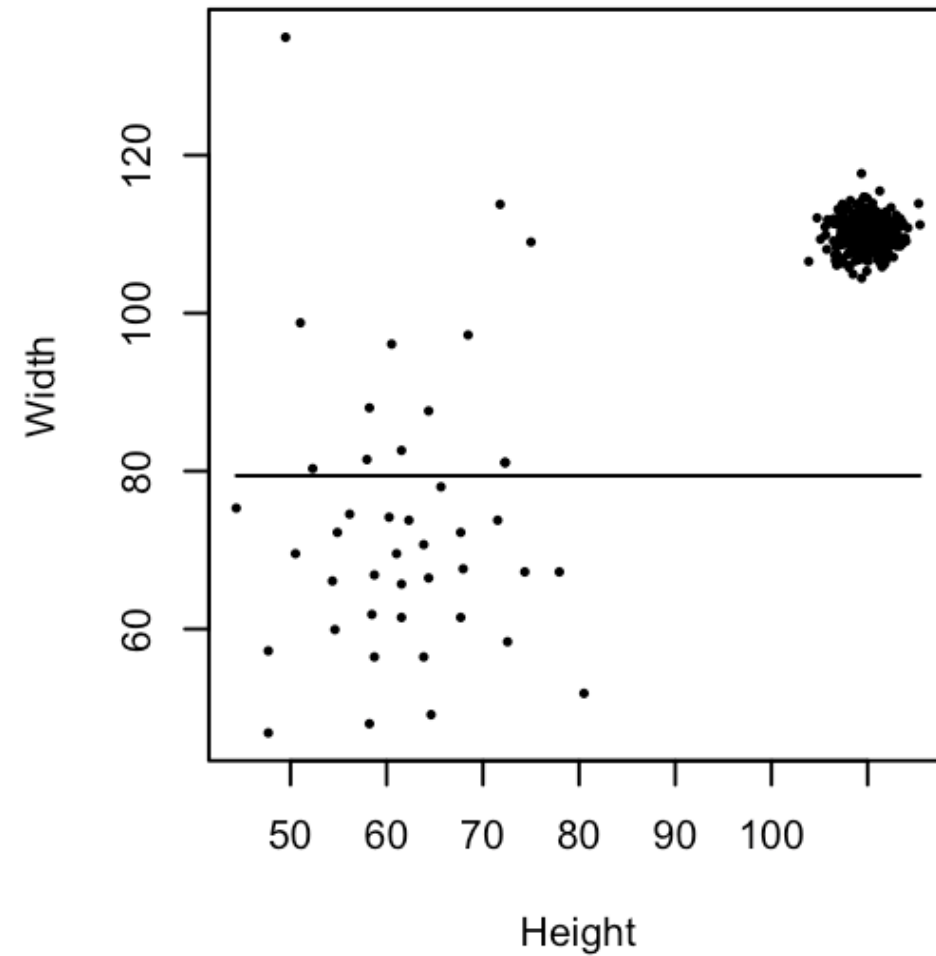
Data Scientist

# Isolation tree

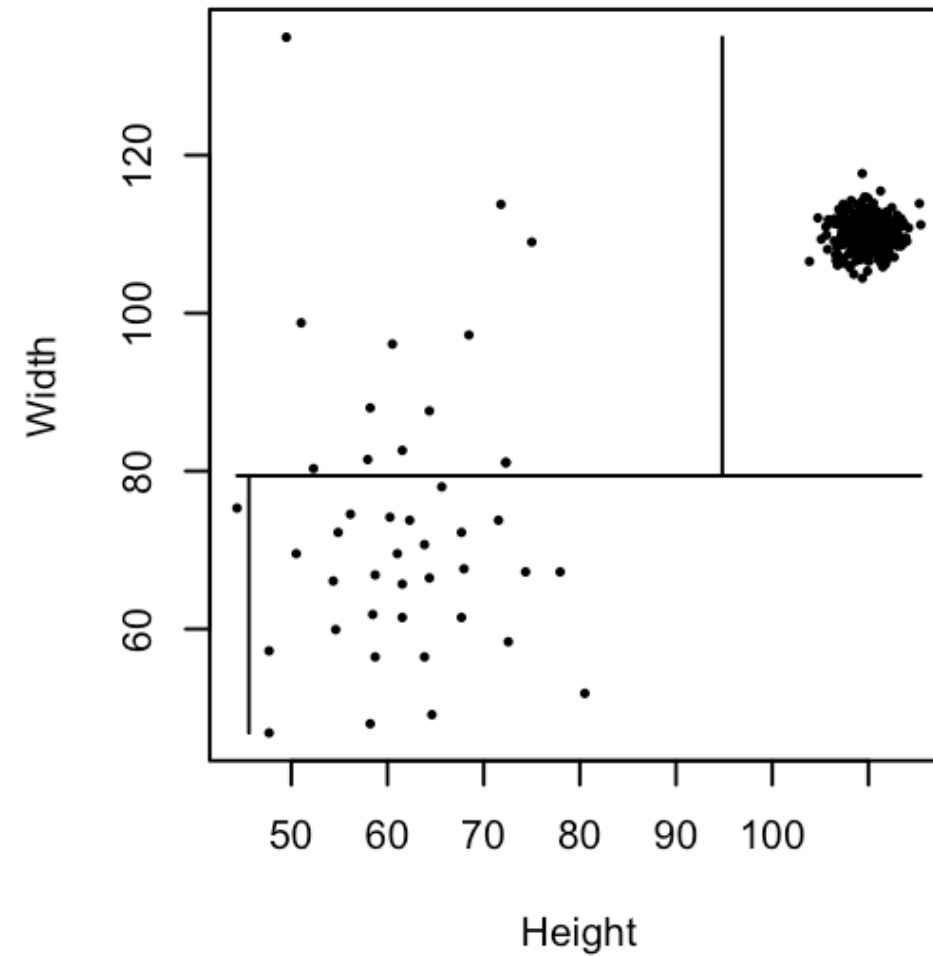


# Isolation tree plots

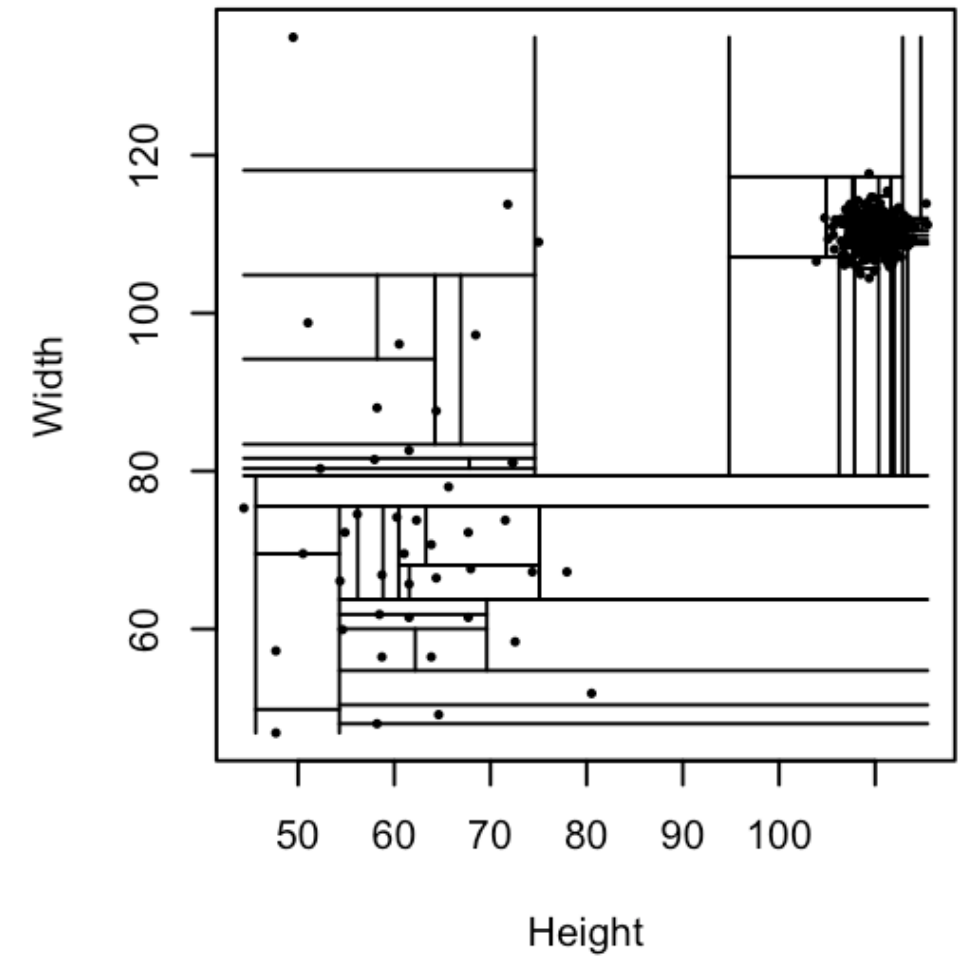
1 Split



3 Splits



All splits



# Fit an isolation tree

```
library(isofor)
furniture_tree <- iForest(data = furniture, nt = 1)
```

## `iForest()` arguments

- `data` - dataframe
- `nt` - number of isolation trees to grow

Package download from <https://github.com/Zelazny7/isofor>

# Generate an isolation score

```
furniture_score <- predict(furniture_tree, newdata = furniture)
```

## `predict()` arguments

- `object` - a fitted `iForest` model
- `newdata` - data to score

# Interpreting the isolation score

```
furniture_score[1:10]
```

```
[1] 0.5820092 0.5820092 0.5439338 0.5820092 0.5439338  
[6] 0.5820092 0.7129862 0.5363547 0.5363547 0.5363547
```

## Standardized path length

- Scores between 0 and 1
- Scores near 1 indicate anomalies (small path length)

**Let's practice!**  
ANOMALY DETECTION IN R

# Isolation forest

ANOMALY DETECTION IN R



**Alastair Rushworth**

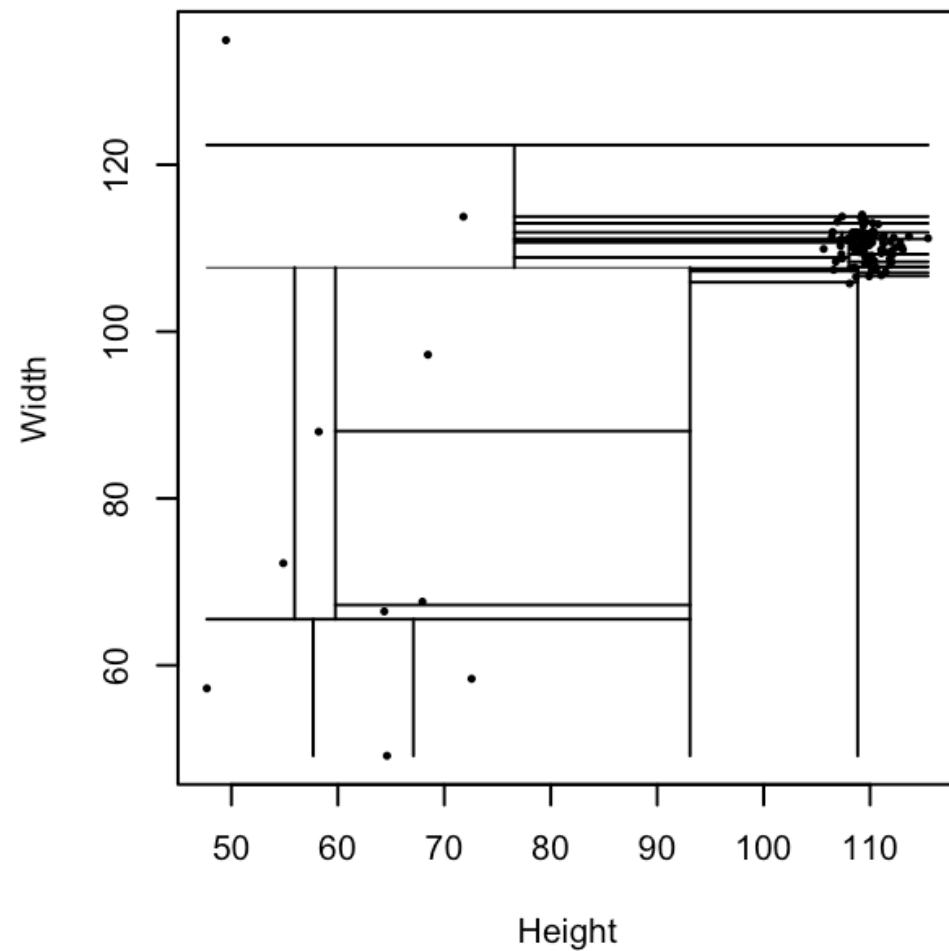
Data Scientist



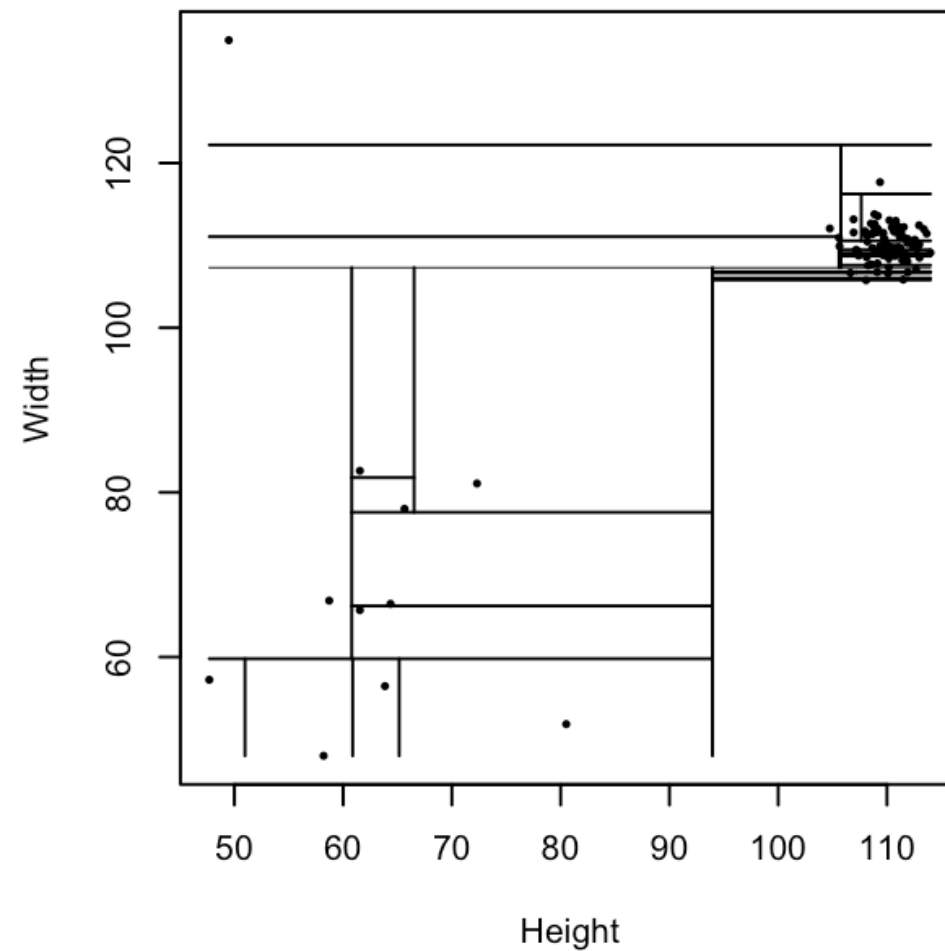
# Sampling to build trees

```
furniture_tree <- iForest(data = furniture, nt = 1, phi = 100)
```

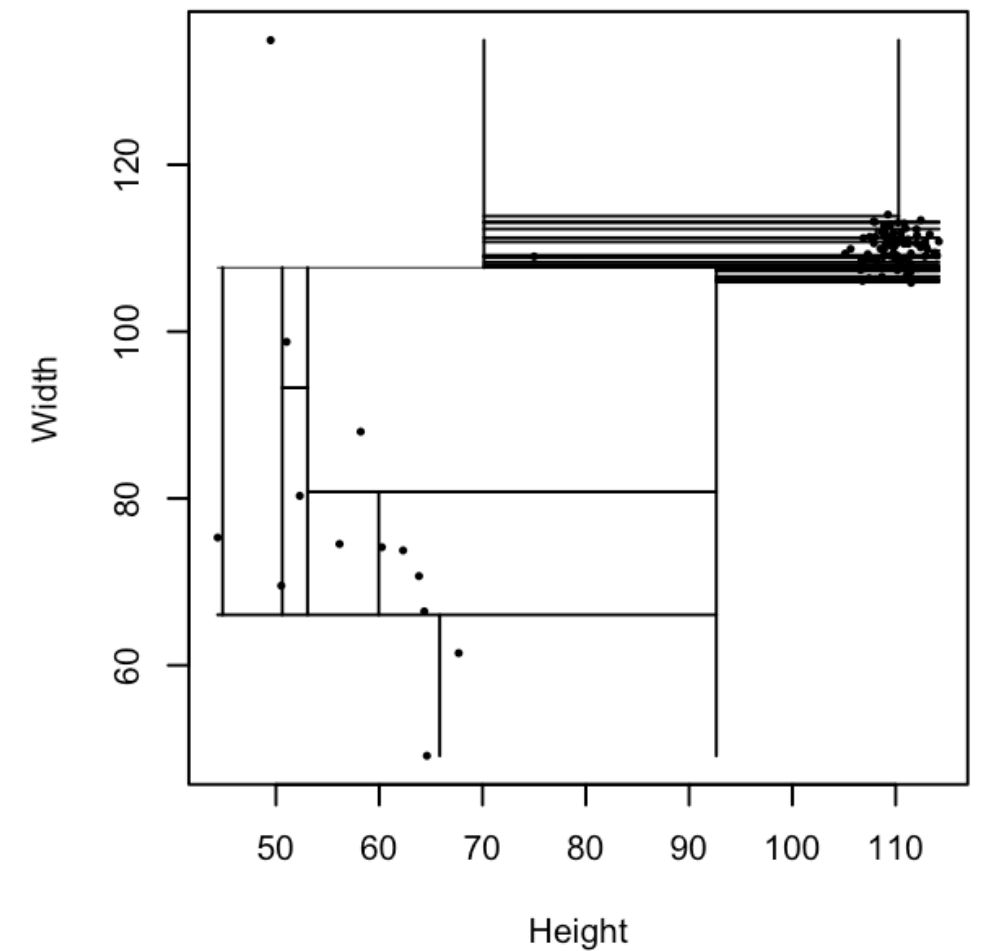
Random tree 1



Random tree 2



Random tree 3



# A forest of many trees

```
furniture_forest <- iForest(data = furniture, nt = 100)
```

## Forest versus single tree

- Average score is robust
- Fast to grow

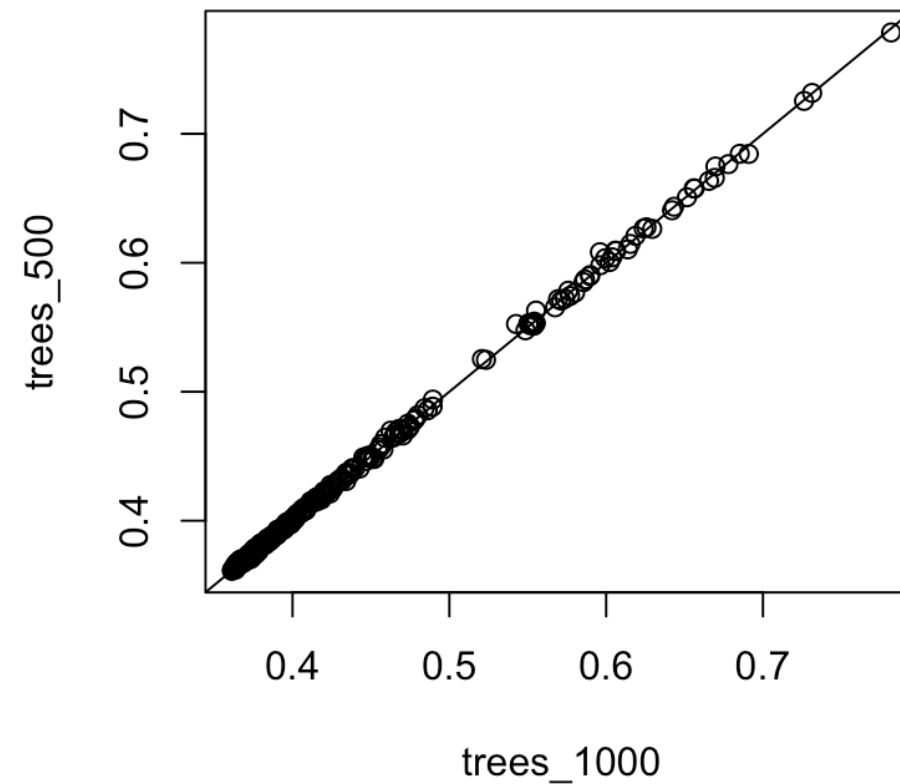
# How many trees?

```
head(furniture_scores)
```

```
   trees_10  trees_50 trees_100 trees_200 trees_500 trees_1000
1 0.5699958 0.5888690 0.5966556 0.5911285 0.6006028 0.6022553
2 0.5930155 0.6094254 0.6102873 0.6067693 0.6103950 0.6138331
3 0.5491612 0.5530659 0.5509151 0.5478388 0.5543705 0.5541810
4 0.5919385 0.5934920 0.6036891 0.5986545 0.6042257 0.6038739
5 0.5755555 0.5545840 0.5562077 0.5502717 0.5529810 0.5533804
6 0.6099932 0.6156158 0.6246391 0.6237609 0.6262847 0.6293865
```

# Score convergence

```
plot(trees_500 ~ trees_1000, data = furniture_scores)  
abline(a = 0, b = 1)
```



**Let's practice!**  
ANOMALY DETECTION IN R

# Visualizing the isolation score

ANOMALY DETECTION IN R



**Alastair Rushworth**  
Data Scientist

# Sequences of values

```
h_seq <- seq(min(furniture$Height), max(furniture$Height), length.out = 20)
```

```
w_seq <- seq(min(furniture$Width), max(furniture$Width), length.out = 20)
```

## `seq()` arguments

- `from` - upper bound
- `to` - lower bound
- `length.out` - values in the sequence

# Building a grid

```
furniture_grid <- expand.grid(Width = w_seq, Height = h_seq)  
head(furniture_grid)
```

```
      Width Height  
1 46.85100 44.359  
2 51.48663 44.359  
3 56.12225 44.359  
4 60.75788 44.359  
5 65.39351 44.359  
6 70.02913 44.359
```

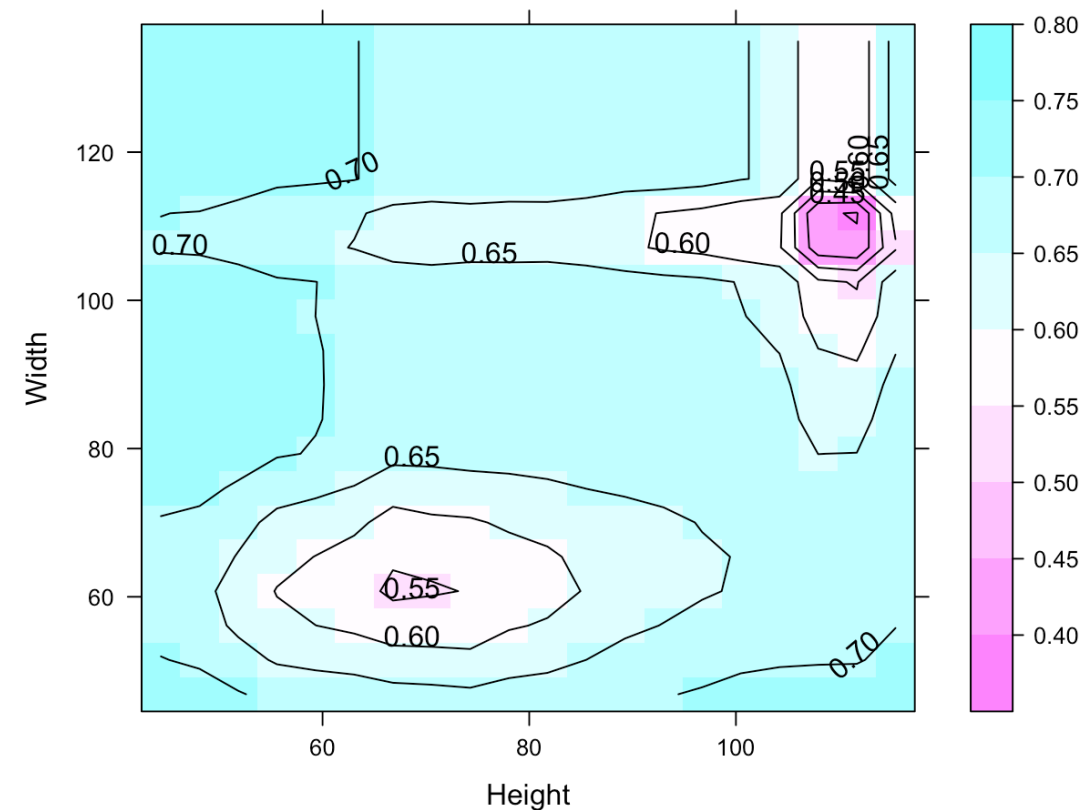


# Scoring the grid

```
furniture_grid$score <- predict(furniture_forest, furniture_grid)
```

# Make the contour plot!

```
library(lattice)
contourplot(score ~ Height + Width, data = furniture_grid,
            region = TRUE)
```



**Let's practice!**  
ANOMALY DETECTION IN R