ANOMALY DETECTION IN R

# Labeled anomalies

Alastair Rushworth
Data Scientist

# Satellite image data

```
head(sat, 5)

  label V1  V2  V3 V4 V5
1     0 92 115 120 94 84
2     0 84 102 106 79 84
3     0 84 102 102 83 80
4     0 80 102 102 79 84
5     0 84  94 102 79 80
```

```
table(sat$label)

   0    1
5732   71
```

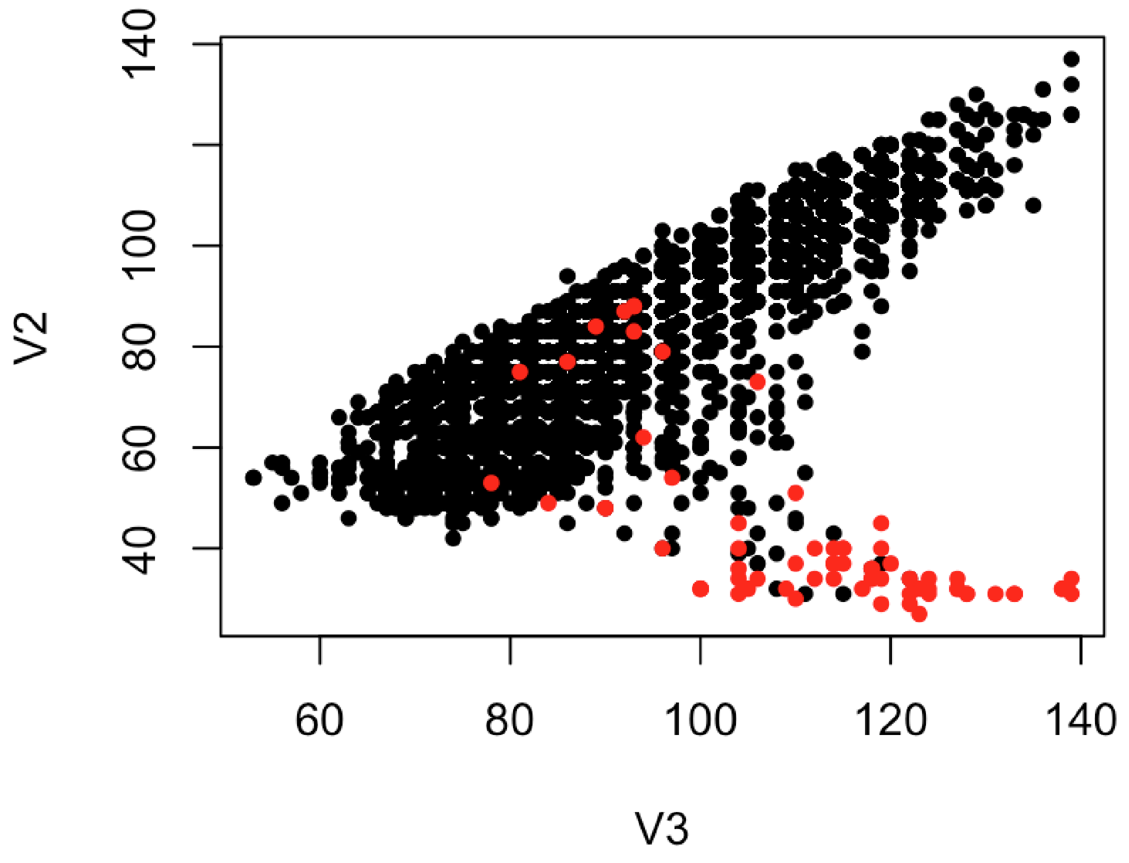
## Cotton crop image proportion

```
71 / 5803

0.01223505
```

# Visualize true anomalies

```
plot(V2 ~ V3, data = sat, col = as.factor(label), pch = 20)
```

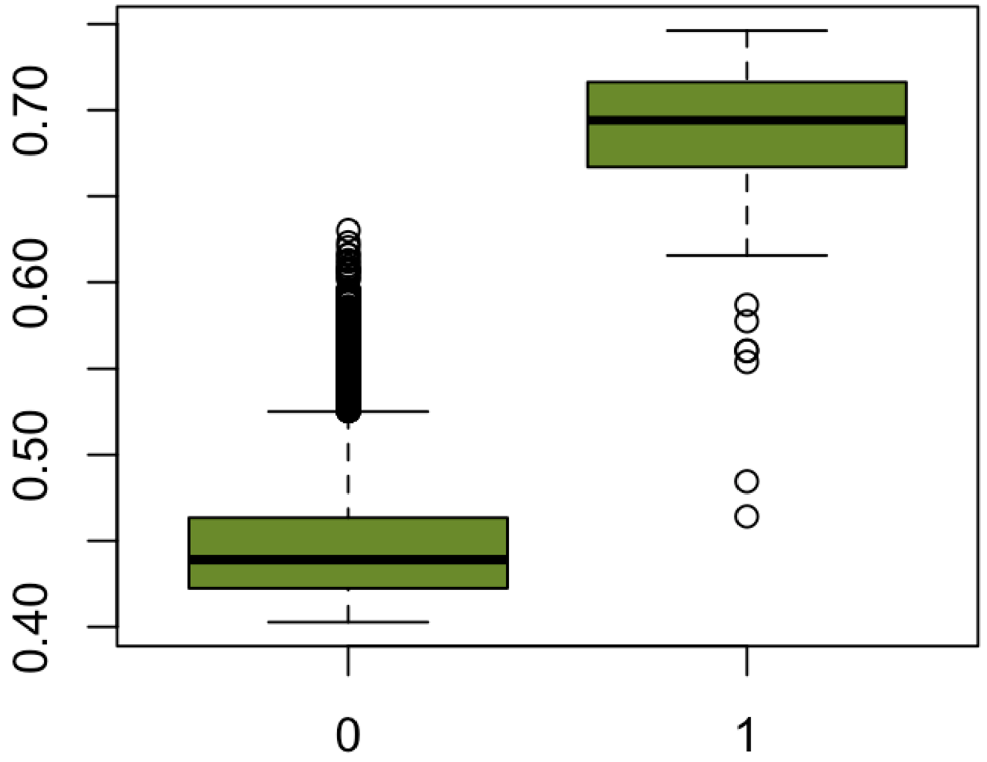


# Anomaly score versus true label

```
sat_for <- iForest(sat[, -1], nt = 100)

sat$score <- predict(sat_for, features)

boxplot(score ~ label, data = sat, col = "olivedrab4")
```



# Why not use models to predict labels?

**Example 1: Detecting rare disease cases**

- Too few cases

**Example 2: Credit card fraud**

- Changes rapidly

ANOMALY DETECTION IN R

# Let's practice!

ANOMALY DETECTION IN R

# Measuring performance

Alastair Rushworth
Data Scientist

# Using a decision threshold

## Choose a high value

```
high_score <- quantile(sat$score, probs = 0.99)
high_score

      99%
0.6228078
```

## Binarize score

```
sat$binary_score <- as.numeric(score >= high_score)
```

# Tables of agreement

Comparing true label and binarized score

```
table(sat$label, sat$binary_score)

        0     1
  0  5729     3
  1    15    56
```

- 56 out of 71 anomalies found

# Recall

Anomalies correctly identified ÷ Total anomalies

- 1 = Perfect recall; every anomaly detected by algorithm

```
table(sat$label, sat$binary_score)

        0    1
  0  5729    3
  1    15   56
```

```
recall <- 56 / (15 + 56)
recall
[1] 0.7887324
```

# Precision

Anomalies correctly identified ÷ Total scored as anomalous

- 1 = Perfect precision; no normal instances incorrectly labeled

```
table(sat$label, sat$binary_score)

        0    1
  0  5729    3
  1    15   56
```

```
precision <- 56 / (56 + 3)
precision

[1] 0.9491525
```

ANOMALY DETECTION IN R

# Let's practice!

ANOMALY DETECTION IN R

# Working with categorical features

Alastair Rushworth
Data Scientist

# Checking column classes

## Class of a single column

```
class(sat$V1)

[1] "numeric"
```

## Class of all columns

```
sapply(X = sat, FUN = class)

      label          V1          V2          V3
  "numeric"   "numeric"   "numeric"   "numeric"
         V4          V5          V6    high_low
  "numeric"   "numeric"   "numeric" "character"
```

# Isolation forest

## Encode categorical features as factor

```
sat$high_low <- as.factor(sat$high_low)
```

```
class(sat$high_low)

[1] "factor"
```

## Train isolation forest

```
sat_for <- iForest(sat[, -1], nt = 100)
```

# LOF with factors

**Gower distance** measures distance between points with categorical & numeric features

```
library(cluster)
sat_dist <- daisy(sat[, -1], metric = "gower")
```

Pass `sat_dist` to `lof`

```
sat_lof <- lof(sat_dist, k = 10)
```

# Exploring Gower distance matrix

- Convert object to matrix

```
sat_distmat <- as.matrix(sat_dist)
```

- Find max and min interpoint distances

```
range(sat_distmat)
```

```
[1] 0.0000000 0.8680774
```

ANOMALY DETECTION IN R

# Let's practice!

ANOMALY DETECTION IN R

# Recap: Anomaly Detection in R

Alastair Rushworth
Data Scientist

# Course summary

**Chapter 1**
Testing and visualizing outliers for single variable and time series

**Chapter 2**
Distance and density based anomaly detection

**Chapter 3**
Tree based anomaly detection

**Chapter 4**
Comparing performance and using factors

# What's next?

- **Model tuning:** eg. choosing $k$ for LOF & kNN
- **Many other techniques:** One-class SVM & clustering approaches

ANOMALY DETECTION IN R

# Congratulations!