# Outlier Detection: Isolation Forest (Penguins)

Darian-Florian Voda

2022-12-30

## Loading packages

```
library(caret)
library(dplyr)
library(mltools)
library(rpart)
library(isotree)
library(plotly)
```

## Exercise

Let's check if there are anomalies in the dataset *"PenguinsWithoutMissingValues.csv"*. We want to use Isolation Forests for it since they have some advantages compared with other anomaly detection techniques.

- import the data PenguinsWithoutMissingValues

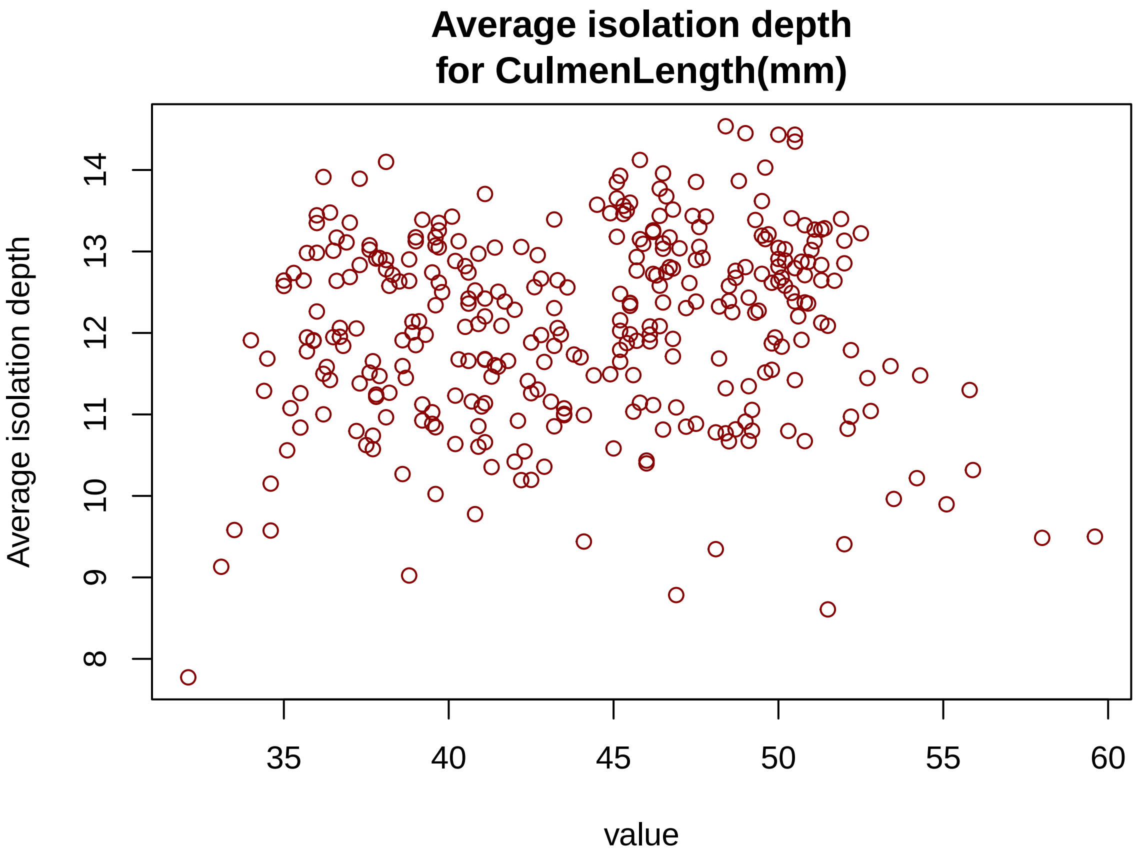
```
# import the data PenguinsWithoutMissingValues
setwd("C:/Users/Dari-Laptop/Desktop/FH Karnten - Master - AppDs/StatisticsAppDSLaptop")
data = read.csv("PenguinsWithoutMissingValues.csv")
```

- encode categorical data

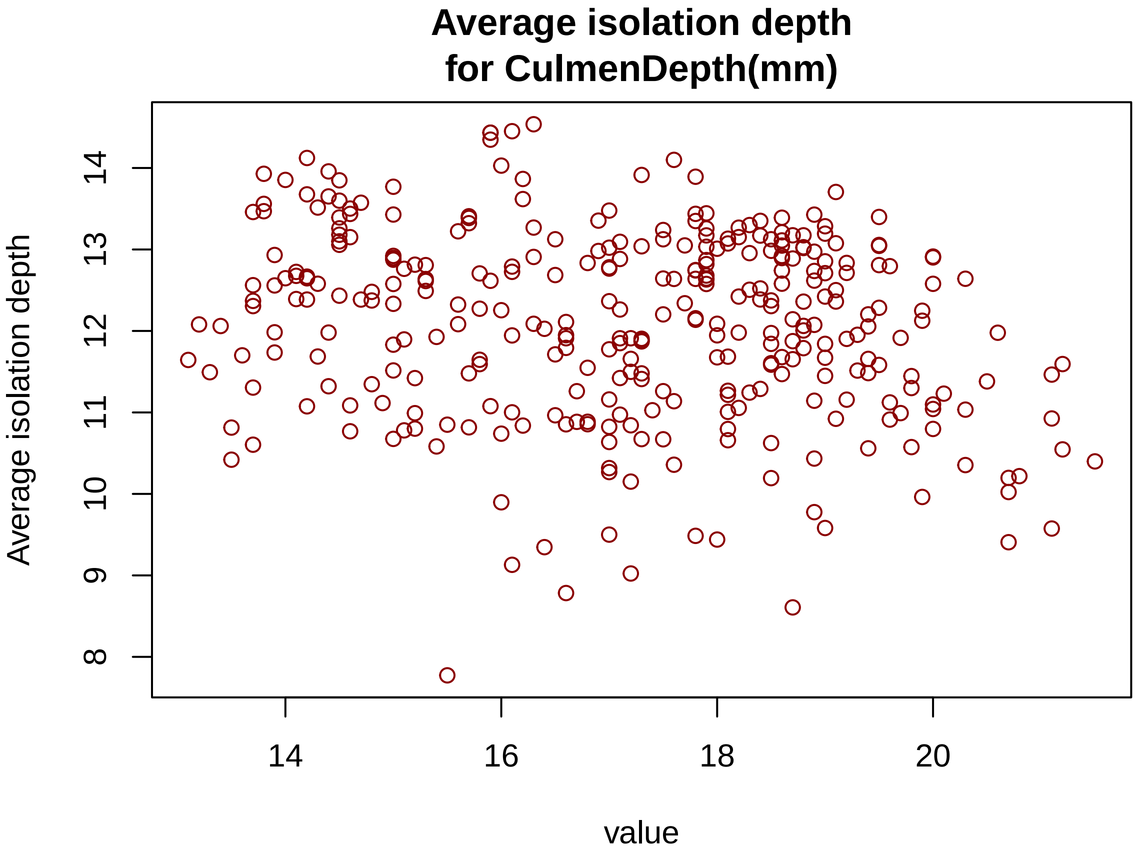
```
# encode categorical data
data$Species = factor(data$Species)
data$Island = factor(data$Island)
data$Gender = factor(data$Gender)
```

- predict the average depth of the isolation forest (using the R library isotree)

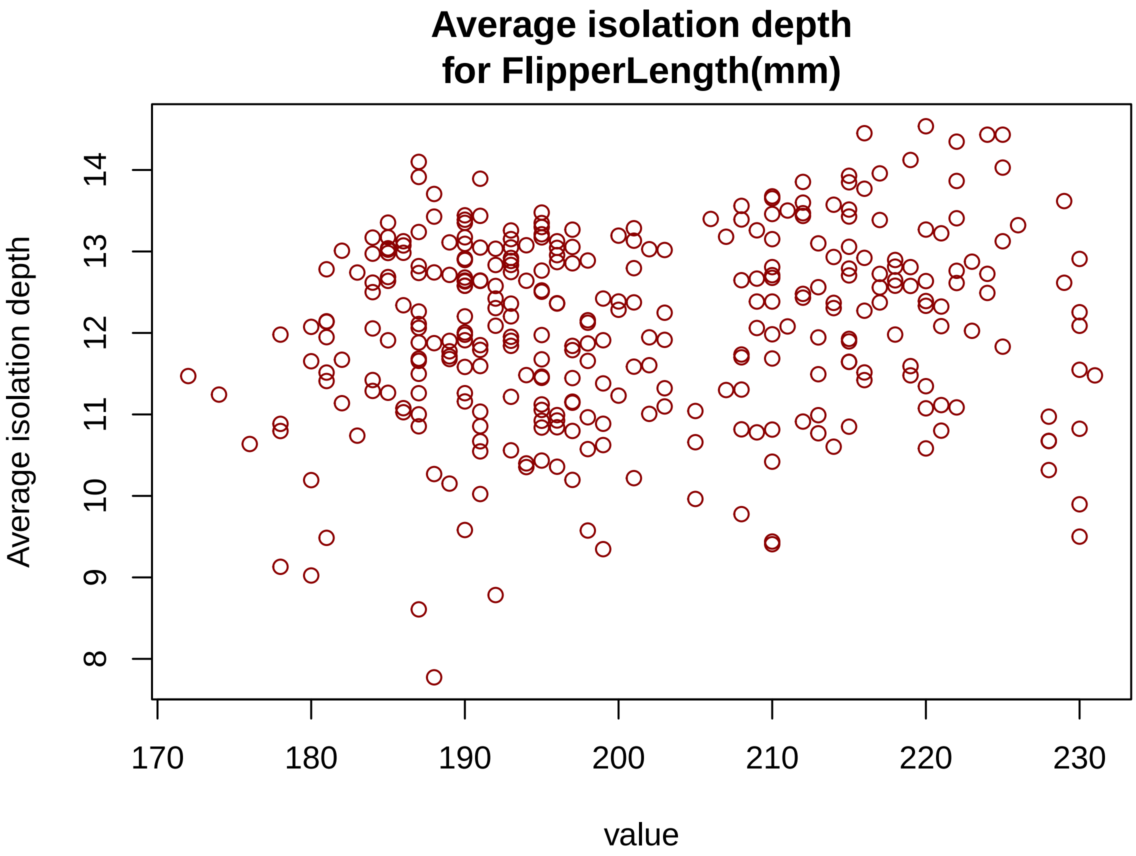
```
# predict the average depth

model <- isolation.forest(data, ndim=1, ntrees=10, nthreads=1)
scores <- predict(model, data, type="avg_depth")
par(mar = c(4,5,3,2))
plot(data$CulmenLength.mm., scores, type="p", col="darkred",
     main="Average isolation depth\nfor CulmenLength(mm)",
     xlab="value", ylab="Average isolation depth")
```



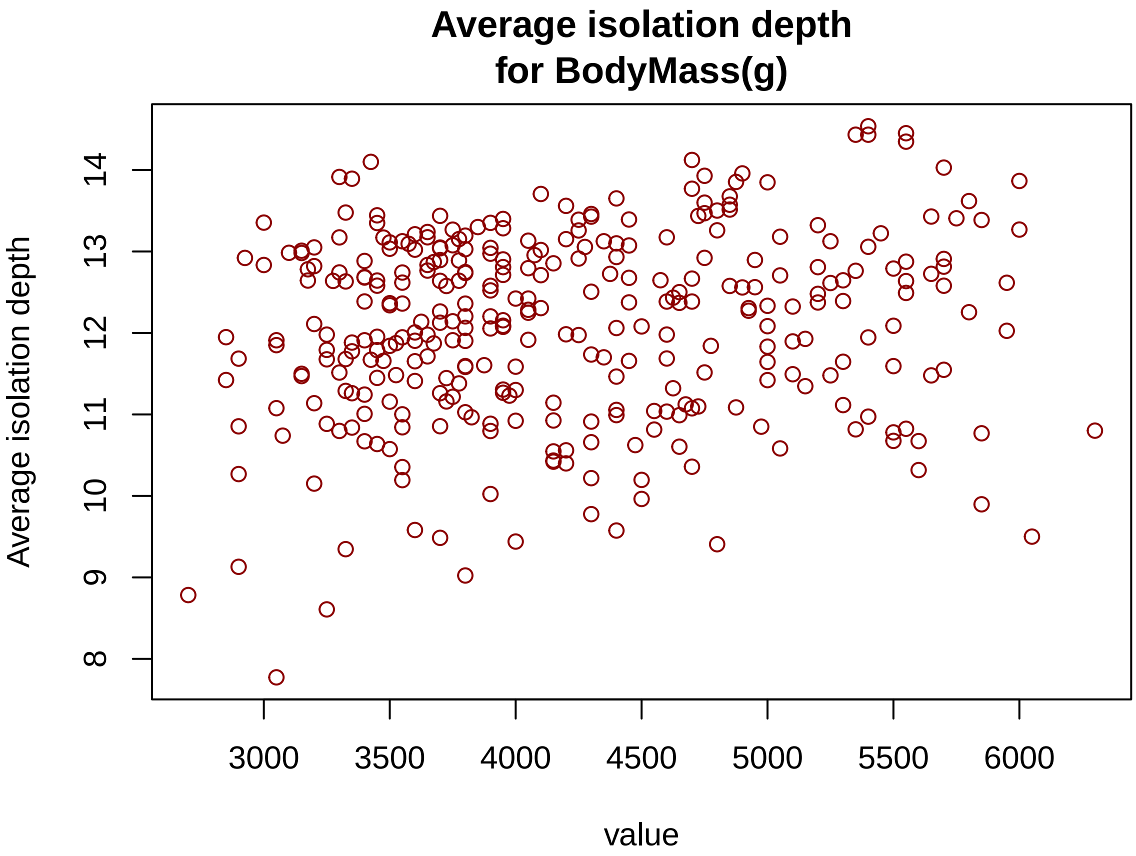
```
plot(data$CulmenDepth.mm., scores, type="p", col="darkred",
     main="Average isolation depth\nfor CulmenDepth(mm)",
     xlab="value", ylab="Average isolation depth")
```



```
plot(data$FlipperLength.mm., scores, type="p", col="darkred",
     main="Average isolation depth\nfor FlipperLength(mm)",
     xlab="value", ylab="Average isolation depth")
```



```
plot(data$BodyMass.g., scores, type="p", col="darkred",
     main="Average isolation depth\nfor BodyMass(g)",
     xlab="value", ylab="Average isolation depth")
```



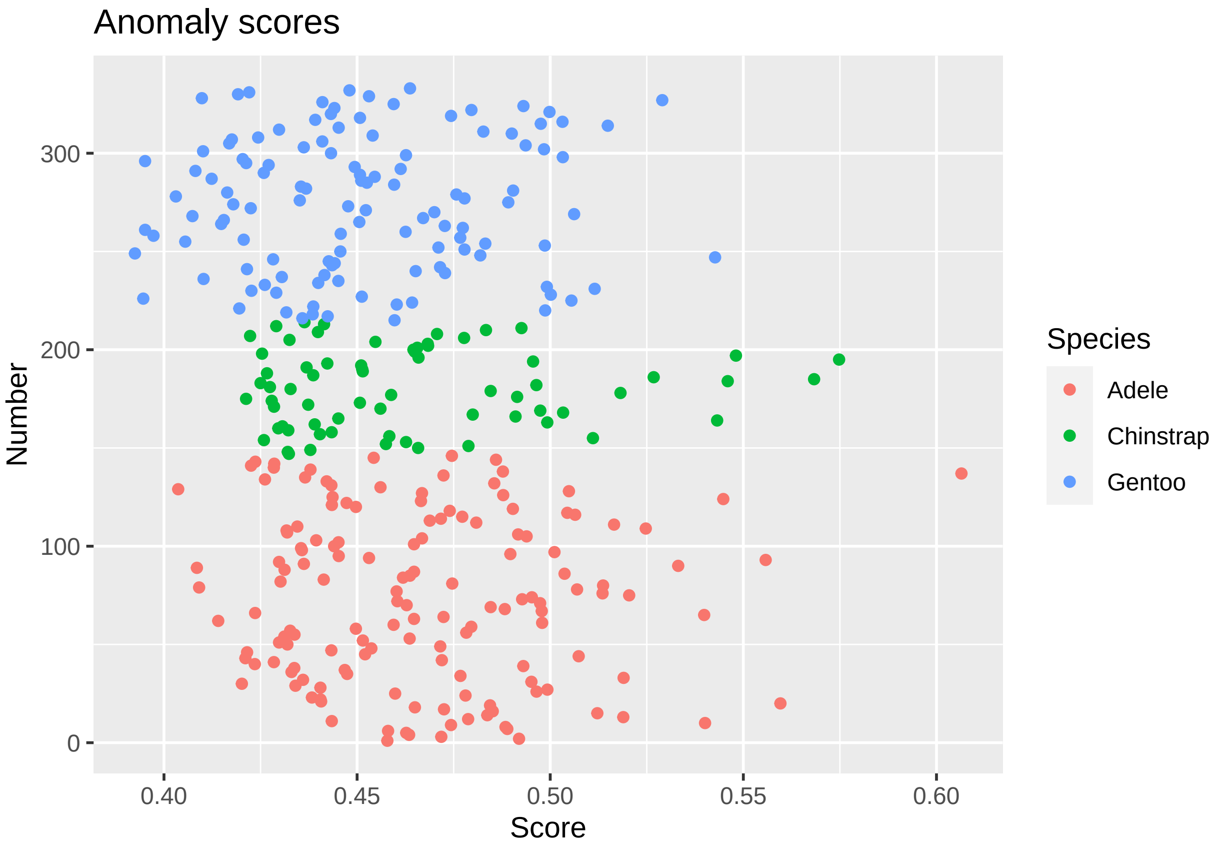
- calculate the anomaly score (see slides or original paper)

```
data$score <- predict(model, newdata = data)

data$score
```

```
##   [1] 0.4578364 0.4919289 0.4717983 0.4634670 0.4627224 0.4580191 0.4888889
##   [8] 0.4884262 0.4743137 0.5400898 0.4434475 0.4787488 0.5189226 0.4837096
##  [15] 0.5121948 0.4851209 0.4725081 0.4649658 0.4844198 0.5596001 0.4406936
##  [22] 0.4405415 0.4383124 0.4780684 0.4598501 0.4964540 0.4992688 0.4405060
##  [29] 0.4340397 0.4201672 0.4951317 0.4360031 0.5190058 0.4767544 0.4474137
##  [36] 0.4330214 0.4468157 0.4337378 0.4930378 0.4235189 0.4284610 0.4719479
##  [43] 0.4210829 0.5073663 0.4520723 0.4215195 0.4433411 0.4537013 0.4715384
##  [50] 0.4319524 0.4297998 0.4514957 0.4636074 0.4311626 0.4337975 0.4782678
##  [57] 0.4326671 0.4496893 0.4795662 0.4594558 0.4979144 0.4140480 0.4647373
##  [64] 0.4723818 0.5398571 0.4235979 0.4978067 0.4882456 0.4845802 0.4628337
##  [71] 0.4973875 0.4604293 0.4927327 0.4952524 0.5204366 0.5135453 0.4602299
##  [78] 0.5069460 0.4090961 0.5136877 0.4746397 0.4301833 0.4413656 0.4618874
##  [85] 0.4636738 0.5037086 0.4647442 0.4312264 0.4085254 0.5331339 0.4362219
##  [92] 0.4297965 0.5557771 0.4530897 0.4452467 0.4896844 0.5011017 0.4357118
##  [99] 0.4354940 0.4440621 0.4647401 0.4451882 0.4394154 0.4668099 0.4938844
## [106] 0.4916919 0.4318985 0.4317359 0.5247426 0.4345292 0.5165276 0.4808505
## [113] 0.4688223 0.4716961 0.4772260 0.5064542 0.5044225 0.4739909 0.4903396
## [120] 0.4496964 0.4434714 0.4472747 0.4665368 0.5448095 0.4436527 0.4878292
## [127] 0.4668022 0.5048415 0.4036845 0.4560588 0.4433422 0.4855180 0.4421209
## [134] 0.4261492 0.4365401 0.4723759 0.6064683 0.4877523 0.4379296 0.4284479
## [141] 0.4225492 0.4285533 0.4236560 0.4859622 0.4543080 0.4745389 0.4323452
## [148] 0.4320515 0.4379103 0.4658032 0.4788382 0.4574619 0.4626743 0.4258793
## [155] 0.5110611 0.4583544 0.4403821 0.4434096 0.4321954 0.4296062 0.4306709
## [162] 0.4390258 0.4992455 0.5432149 0.4451476 0.4910207 0.4799401 0.5033391
## [169] 0.4974272 0.4560530 0.4284908 0.4373445 0.4507294 0.4279020 0.4212556
## [176] 0.4914383 0.4588224 0.5181932 0.4845815 0.4327766 0.4274348 0.4964224
## [183] 0.4250045 0.5459582 0.5683148 0.5267883 0.4386547 0.4266698 0.4514697
## [190] 0.4513407 0.4369203 0.4510576 0.4422928 0.4955650 0.5747865 0.4659029
## [197] 0.5480835 0.4254013 0.4649305 0.4645948 0.4655877 0.4683809 0.4682885
## [204] 0.4547470 0.4325063 0.4777032 0.4222985 0.4706933 0.4398649 0.4833682
## [211] 0.4925572 0.4290702 0.4414542 0.4363800 0.4597009 0.4358379 0.4423925
## [218] 0.4385070 0.4316842 0.4987000 0.4195007 0.4386651 0.4602693 0.4642384
## [225] 0.5054918 0.3946461 0.4512227 0.5001638 0.4290825 0.4226430 0.5115130
## [232] 0.4991299 0.4261074 0.4399429 0.4451592 0.4102558 0.4305178 0.4415535
## [239] 0.4727690 0.4651656 0.4214929 0.4714859 0.4435263 0.4442092 0.4426693
## [246] 0.4282673 0.5426833 0.4819088 0.3924836 0.4456695 0.4778104 0.4710584
## [253] 0.4986112 0.4831991 0.4055162 0.4206496 0.4767074 0.3972894 0.4457665
## [260] 0.4625595 0.3951124 0.4773757 0.4727027 0.4148229 0.4505772 0.4155020
## [267] 0.4670992 0.4073763 0.5061868 0.4700047 0.4522723 0.4224750 0.4476955
## [274] 0.4179520 0.4891470 0.4351705 0.4778159 0.4030806 0.4757101 0.4163724
## [281] 0.4904007 0.4367893 0.4354727 0.4595978 0.4525370 0.4510837 0.4123429
## [288] 0.4545669 0.4507523 0.4258496 0.4081464 0.4612861 0.4493849 0.4271062
## [295] 0.4212851 0.3951124 0.4203855 0.5032602 0.4626600 0.4432540 0.4101284
## [302] 0.4983920 0.4362031 0.4936398 0.4168934 0.4409982 0.4175788 0.4243844
## [309] 0.4540306 0.4900495 0.4826956 0.4297941 0.4452357 0.5149059 0.4975624
## [316] 0.5031785 0.4391660 0.4507518 0.4743276 0.4432246 0.4998094 0.4796001
## [323] 0.4441266 0.4930681 0.4594674 0.4410230 0.5290099 0.4098169 0.4530819
## [330] 0.4191695 0.4220659 0.4480337 0.4637000
```

- use ggplotly scatterplot to visualize the anomaly scores and show the IndividualIDs in the hoover text

```
# use ggplotly scatterplot to visualize the anomaly scores
# and show the IndividualIDs in the hoover text

p = ggplot(data,
           aes(x=score, y = 1:nrow(data), color=Species,
               text = paste("IndividualID :", IndividualID))) +
  geom_point() +
  labs(title="Anomaly scores",
                      x ="Score", y = "Number")

p
```



```
ggplotly(p)
```

### Conclusions

- Yes, there are anomalies regarding to our anomaly scores
- I have computed the anomaly score using the predict function and not from scratch, even though several trials occured using the anomaly formula from the Isolation Forests article
- ID number N81A1 seems to be an anomaly from its species
- Same applies for ID N88A1, N72A1 and many others as we can see from our plot