What is R?

Statistics Package like SPSS, Minitab, etc. but R is free and opensource

Can be used as command line application (no GUI) or as programming language

Some language features:

- Functional
- Dynamic typed
- Call-by-value
- Built in support for matrices, complex numbers
- 4(?) different approaches for OOP

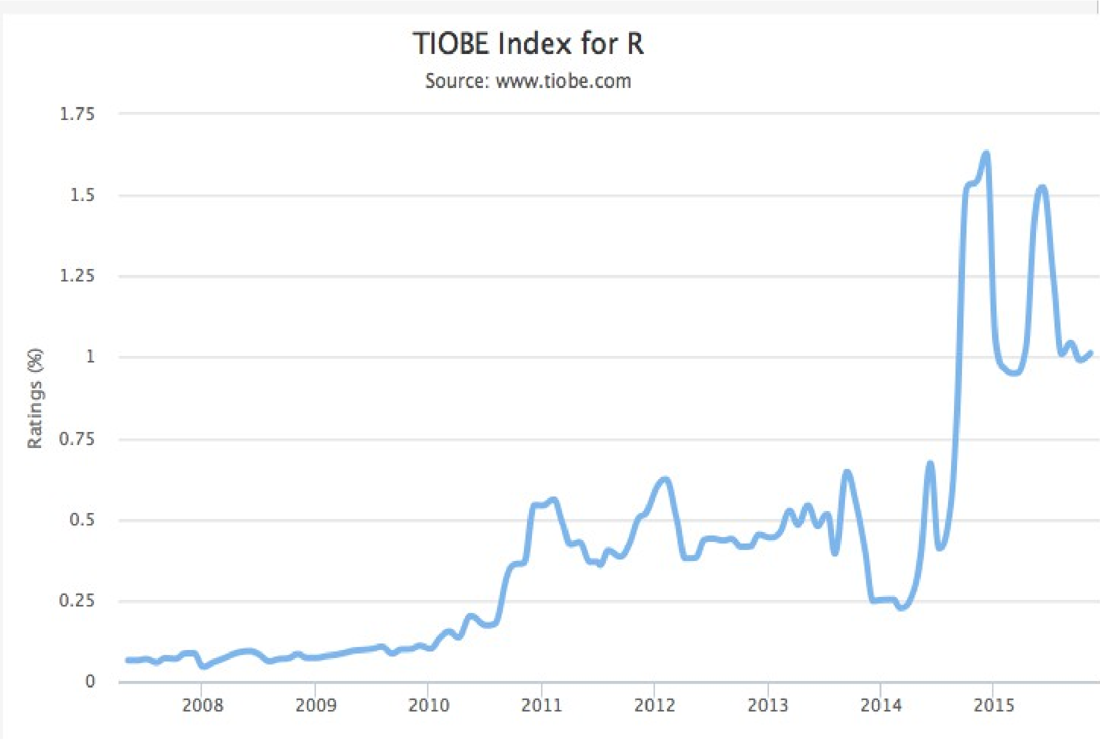
Why we should improve R support

- R is taking over in science in general

- Lots of bioinformatics and a substantial amount of image analysis packages available, e. g. https://cran.r-project.org/web/views/MedicalImaging.html

- Scripts can run on client and server side

- Easy, straightforward using the Java Gateway and rJava



Why we should not

- There is already an indirect way to interact with OMERO by directly accessing the files on the server and using RBioformats package (https://github.com/aoles/RBioFormats) (has not been updated since 5.0.8)
- Maybe there is no demand?

  Email from 2011 and a Trello card from 2013 for R support in OMERO

Examples and “R toolbox”

- Add examples to documentation for using the Java Gateway via rJava
- Build a “toolbox” of methods to make the interaction with R and OMERO more convenient → Needs more external input, use cases, etc.
- Create a Dockerfile building on top of r-base:latest

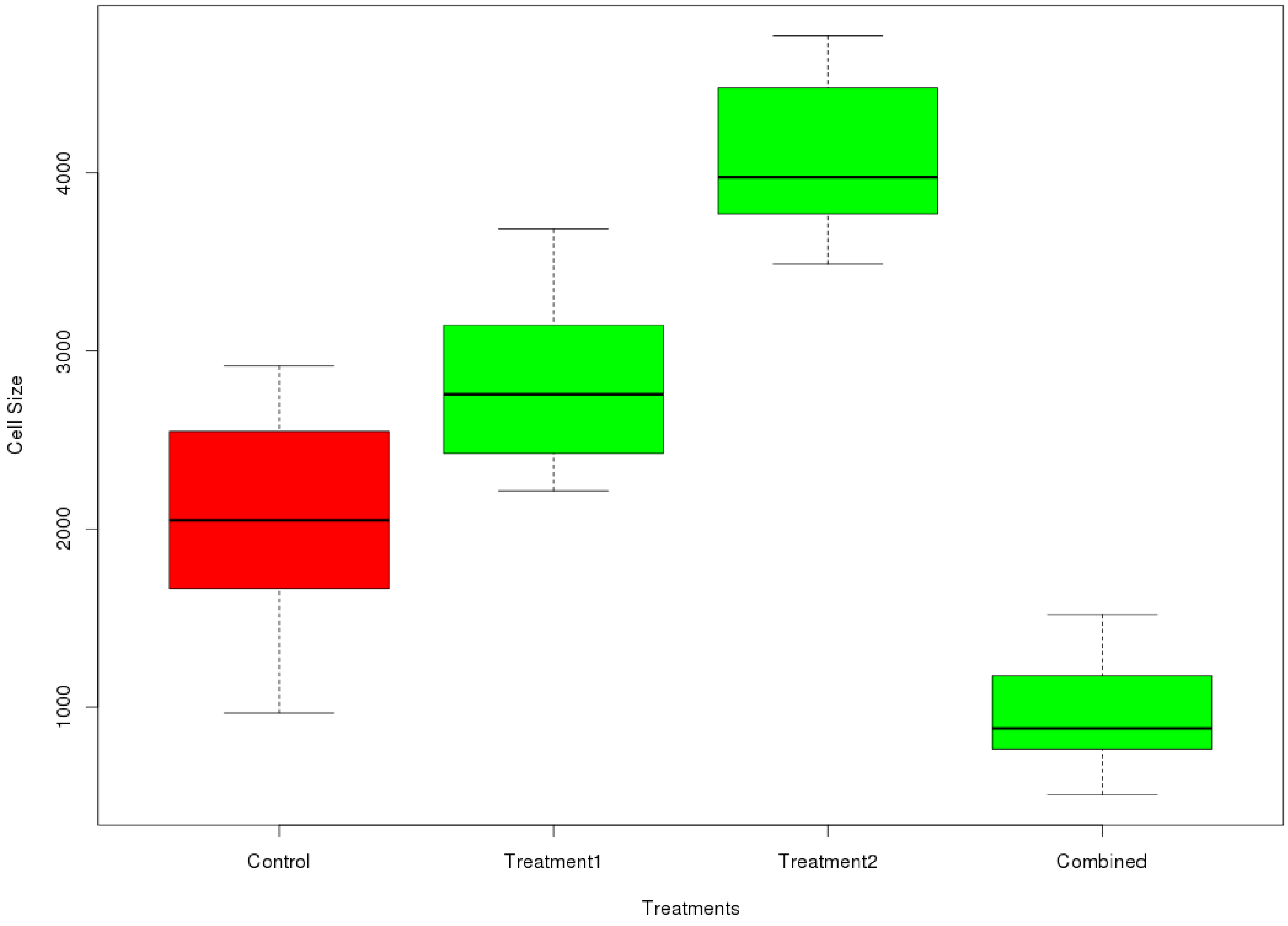
https://github.com/dominikl/rOMERO

Example – Connect to OMERO and build data frame

```r
source("R/gateway.R")

# Connect to OMERO
connect("user", "test", "localhost")

# The dataset we're dealing with and the name of tags
datasetId <- 101
treatment1Tag <- "treatment1"
treatment2Tag <- "treatment2"

# Define the data frame
df <- data.frame(Treatment1 = logical(), Treatment2 = logical(), CellSize = numeric())

# Loop over the images and gather the data
for(image in images) {
  annos <- listImageAnnotations(image)
  rois <- listROIs(image)

  treatment1 <- FALSE
  treatment2 <- FALSE

  for(anno in annos) {
    if(.jinstanceof(anno, "omero/gateway/model/TagAnnotationData")) {
      tag <- .jcast(anno, "omero/gateway/model/TagAnnotationData")
      value <- tag$getTagValue()
      if(value == treatment1Tag)
        treatment1 <- TRUE
      if(value == treatment2Tag)
        treatment2 <- TRUE
    }
  }

  for (roi in rois) {
    row <- c(treatment1, treatment2, as.numeric(roi$getArea()))
    df[nrow(df) + 1, ] <- row
  }
}

# Disconnect again
disconnect()
```

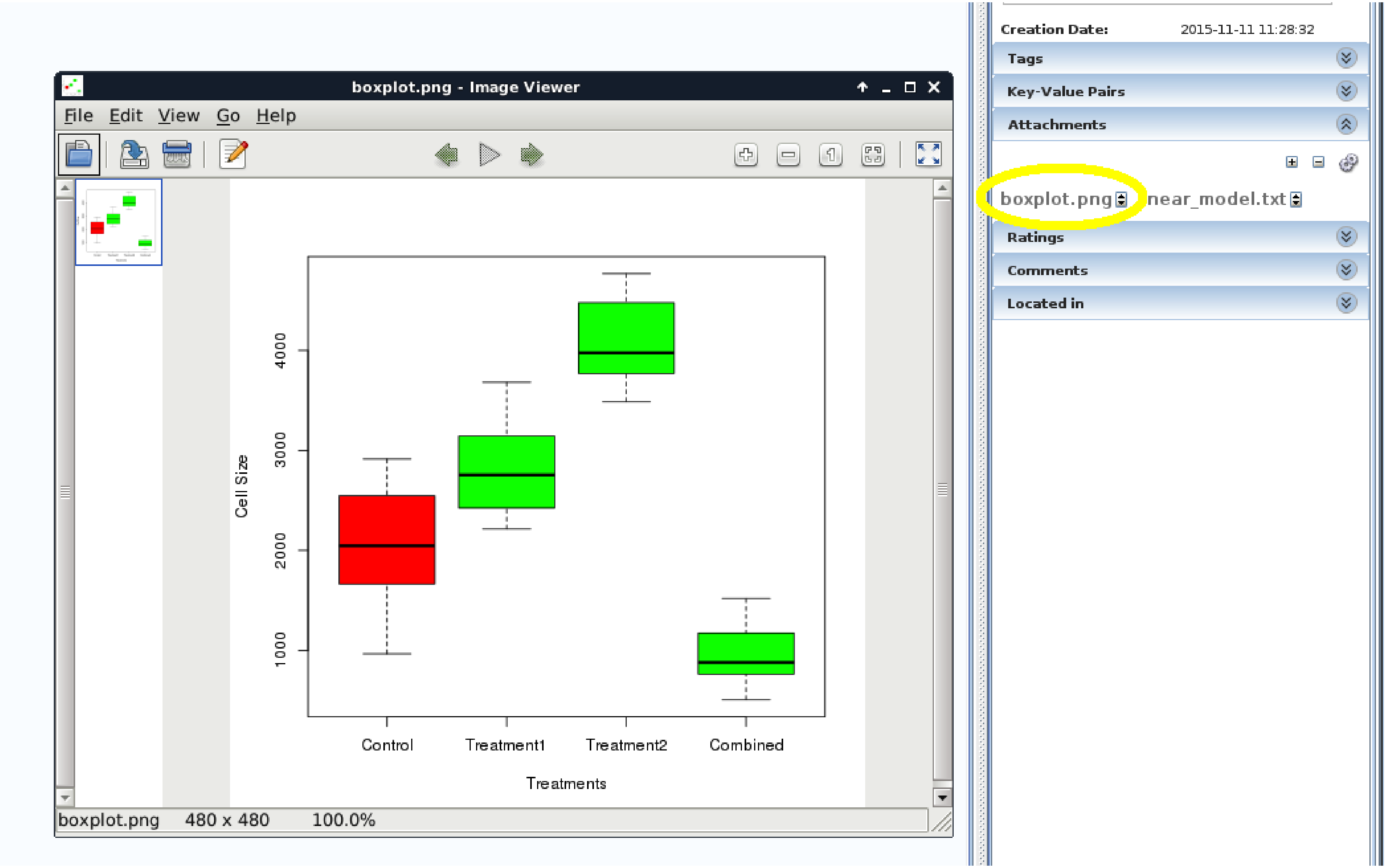
Example – Plot data

```
# Plot the data for a brief overview
boxplot(df$CellSize~df$Treatment1+df$Treatment2, ylab = "Cell Size", xlab ="Treatments",
        names = c("Control", "Treatment1", "Treatment2", "Combined"),
        col = c("red", "green", "green", "green"))
```



Example - Attach plot to OMERO Dataset

```
tmpfile <- "/tmp/boxplot.png"
png(tmpfile)
boxplot(df$CellSize~df$Treatment1+df$Treatment2, ylab = "Cell Size", xlab ="Treatments",
        names = c("Control", "Treatment1", "Treatment2", "Combined"),
        col = c("red", "green", "green", "green"))
dev.off()
addFile(tmpfile, dataset)
```



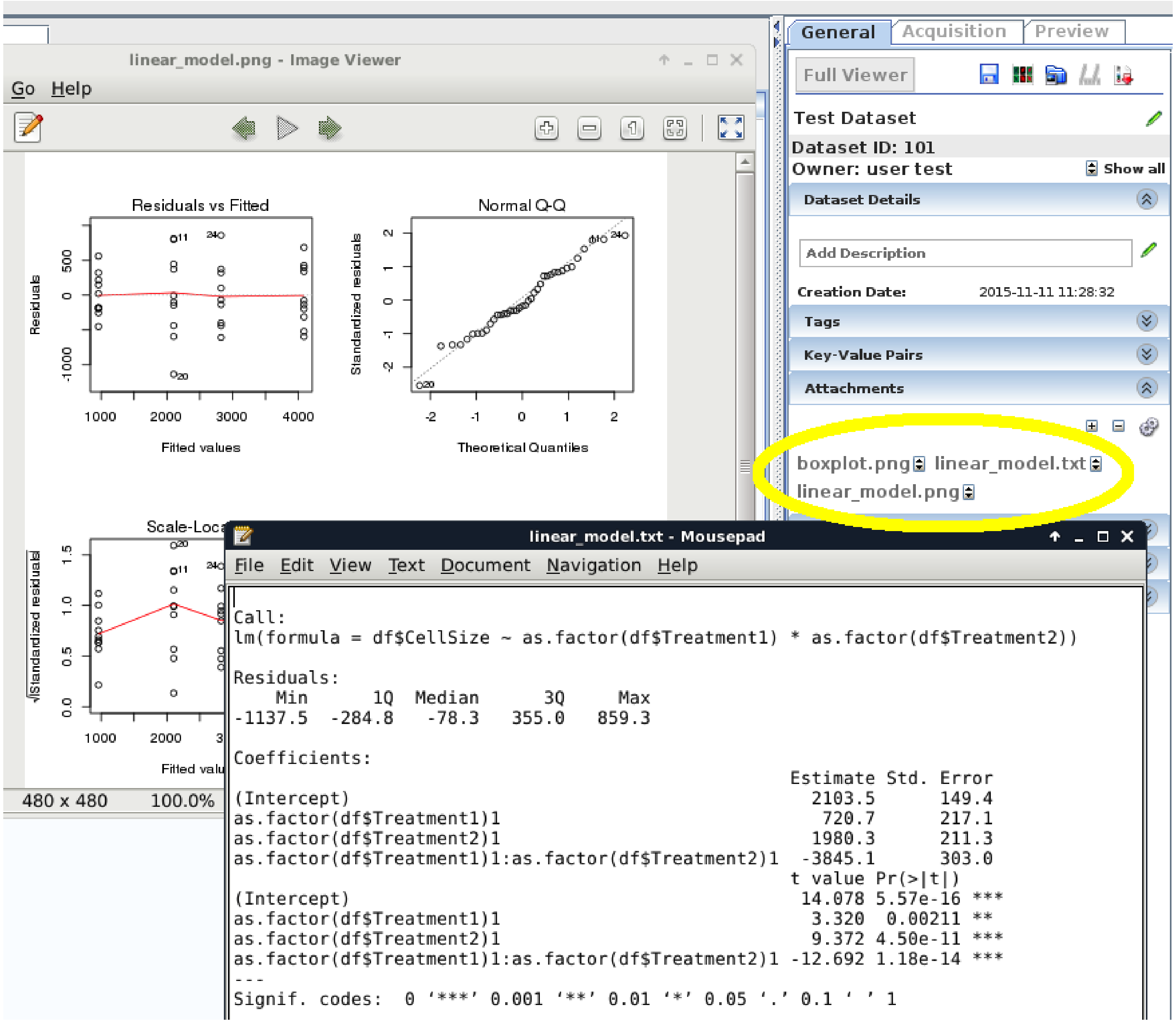
Creation Date: 2015-11-11 11:28:32

boxplot.png near_model.txt

boxplot.png 480 x 480 100.0%

Example – Build model and attach to OMERO Dataset

```
# Build a model
model <- lm(df$CellSize~as.factor(df$Treatment1)*as.factor(df$Treatment2))

# Attach the results to the dataset
tmpfile <- "/tmp/linear_model.txt"
sink(tmpfile)
summary(model)
sink()
addFile(tmpfile, dataset)

tmpfile <- "/tmp/linear_model.png"
png(tmpfile)
# Display four plots in one
par(mfrow=c(2,2))
plot(model)
dev.off()
addFile(tmpfile, dataset)
```



Test Dataset

Dataset ID: 101

Owner: user test

Creation Date: 2015-11-11 11:28:32

```
Call:
lm(formula = df$CellSize ~ as.factor(df$Treatment1) * as.factor(df$Treatment2))

Residuals:
    Min      1Q  Median      3Q     Max
-1137.5  -284.8   -78.3   355.0   859.3

Coefficients:
                                                   Estimate Std. Error
(Intercept)                                          2103.5      149.4
as.factor(df$Treatment1)1                             720.7      217.1
as.factor(df$Treatment2)1                            1980.3      211.3
as.factor(df$Treatment1)1:as.factor(df$Treatment2)1 -3845.1      303.0
                                                   t value Pr(>|t|)
(Intercept)                                          14.078 5.57e-16 ***
as.factor(df$Treatment1)1                             3.320  0.00211 **
as.factor(df$Treatment2)1                             9.372 4.50e-11 ***
as.factor(df$Treatment1)1:as.factor(df$Treatment2)1 -12.692 1.18e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```