Exploring Nondeterminism in Tools That Automatically Generate or Analyse Test Suites

Beatriz Souza

5/14/2019

Installing and loading packages

```
install.packages("ggpubr")
```

```
## Installing package into '/home/beatriz/R/x86_64-pc-linux-gnu-library/3.4'
## (as 'lib' is unspecified)
```

```
install.packages("ggplot2")
```

```
## Installing package into '/home/beatriz/R/x86_64-pc-linux-gnu-library/3.4'
## (as 'lib' is unspecified)
```

```
install.packages("reshape")
```

```
## Installing package into '/home/beatriz/R/x86_64-pc-linux-gnu-library/3.4'
## (as 'lib' is unspecified)
```

```
install.packages("reshape2")
```

```
## Installing package into '/home/beatriz/R/x86_64-pc-linux-gnu-library/3.4'
## (as 'lib' is unspecified)
```

```
install.packages("dplyr")
```

```
## Installing package into '/home/beatriz/R/x86_64-pc-linux-gnu-library/3.4'
## (as 'lib' is unspecified)
```

```
library("ggplot2")
library("reshape")
library("reshape2")
```

```
##
## Attaching package: 'reshape2'
## The following objects are masked from 'package:reshape':
##
##     colsplit, melt, recast
```

```
library("ggpubr")
```

```
## Loading required package: magrittr
```

```
library("dplyr")
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:reshape':
##
##     rename
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

Reading data

```
evo <- read.csv('/home/beatriz/Documents/SRC-2019/statistics/exploringNondeterminism/
evo1Execution.csv', sep = ",", header = TRUE)
evoMelted <- melt(evo, id = c("Project", "Tool"))
#evoMelted

randoop <- read.csv('/home/beatriz/Documents/SRC-2019/statistics/exploringNondetermin
ism/randoop1Execution.csv', sep = ",", header = TRUE)
randoopMelted <- melt(randoop, id = c("Project", "Tool"))
#randoopMelted

pitEvo <- read.csv('/home/beatriz/Documents/SRC-2019/statistics/exploringNondetermini
sm/pitSameProjectSameTestSuiteEvo.csv', sep = ",", header = TRUE)
pitEvoMelted <- melt(pitEvo, id = c("Project", "Tool"))
#pitEvoMelted

pitRandoop <- read.csv('/home/beatriz/Documents/SRC-2019/statistics/exploringNondeter
minism/pitSameProjectSameTestSuiteRandoop.csv', sep = ",", header = TRUE)
pitRandoopMelted <- melt(pitRandoop, id = c("Project", "Tool"))
#pitRandoopMelted

pitManual <- read.csv('/home/beatriz/Documents/SRC-2019/statistics/exploringNondeterm
inism/pitSameProjectSameTestSuiteManual.csv', sep = ",", header = TRUE)
pitManualMelted <- melt(pitManual, id = c("Project", "Tool"))
#pitManualMelted
```

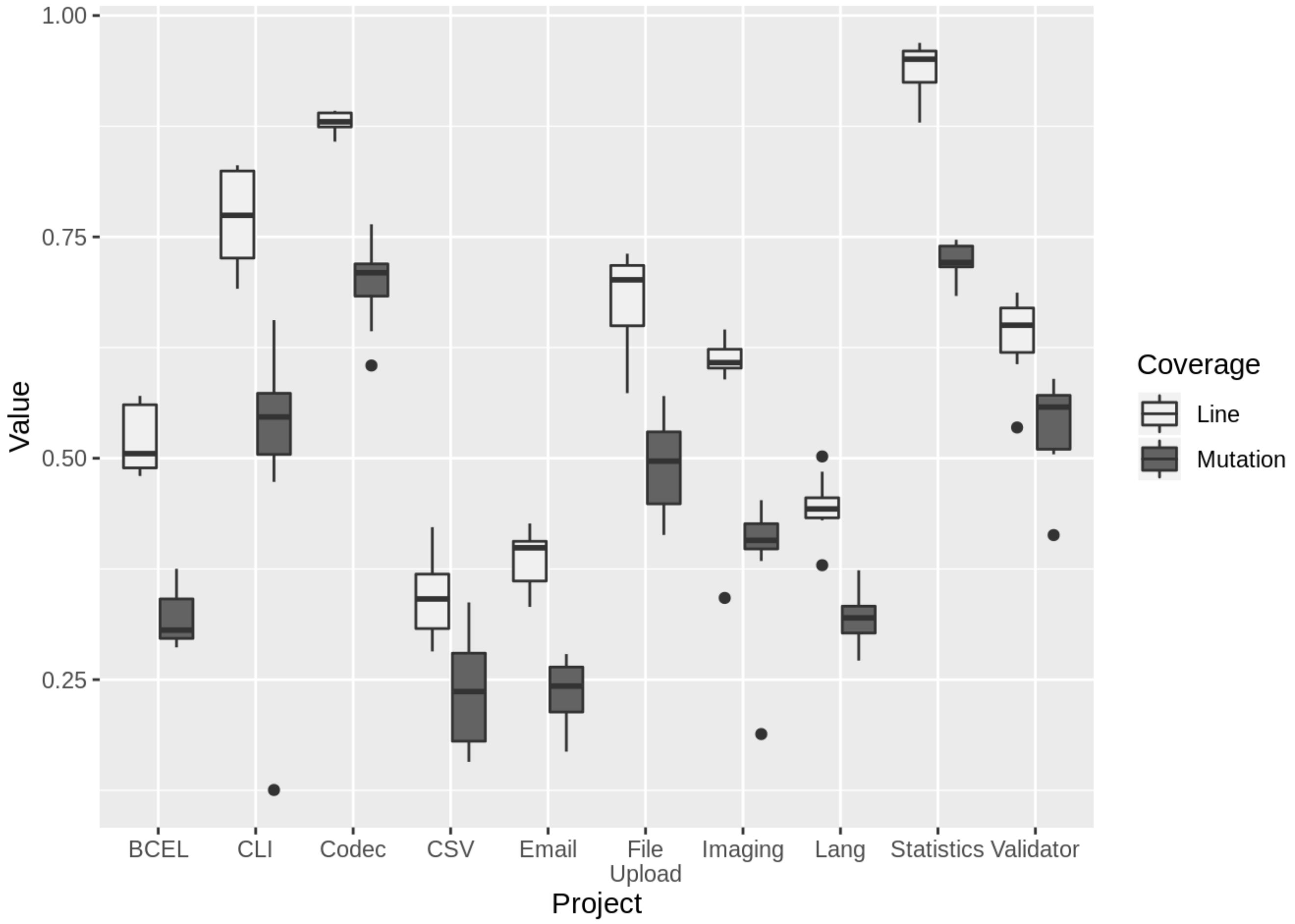
Declaring common values

```
projects=c("BCEL","CLI","Codec", "CSV", "Email", "File\nUpload", "Imaging", "Lang",
"Statistics", "Validator")
```

PIT was executed once for each test suite.

Here we analyse the difference between test suites generated for the same project with EvoSuite in terms of line and mutation coverage.

```
ggplot(data = evoMelted, aes(x=Project, y=value)) +
  geom_boxplot(aes(fill=variable)) +
  scale_x_discrete(labels=projects) +
  labs(x ="Project", y = "Value") +
  scale_fill_manual(values = c("#f0f0f0", "#636363"), name = "Coverage",
                    labels = c("Line", "Mutation"))
```



```
# Line Coverage mean, variation and standard deviation
evoMeanVariationSDLC <- evo %>% group_by(Project) %>% summarise(LineCoverageAverage =
mean(LineCoverage),
                                                              LineCoverageVariation =
var(LineCoverage),
                                                              LineCoverageStandardDev
iation = sd(LineCoverage))
evoMeanVariationSDLC
```

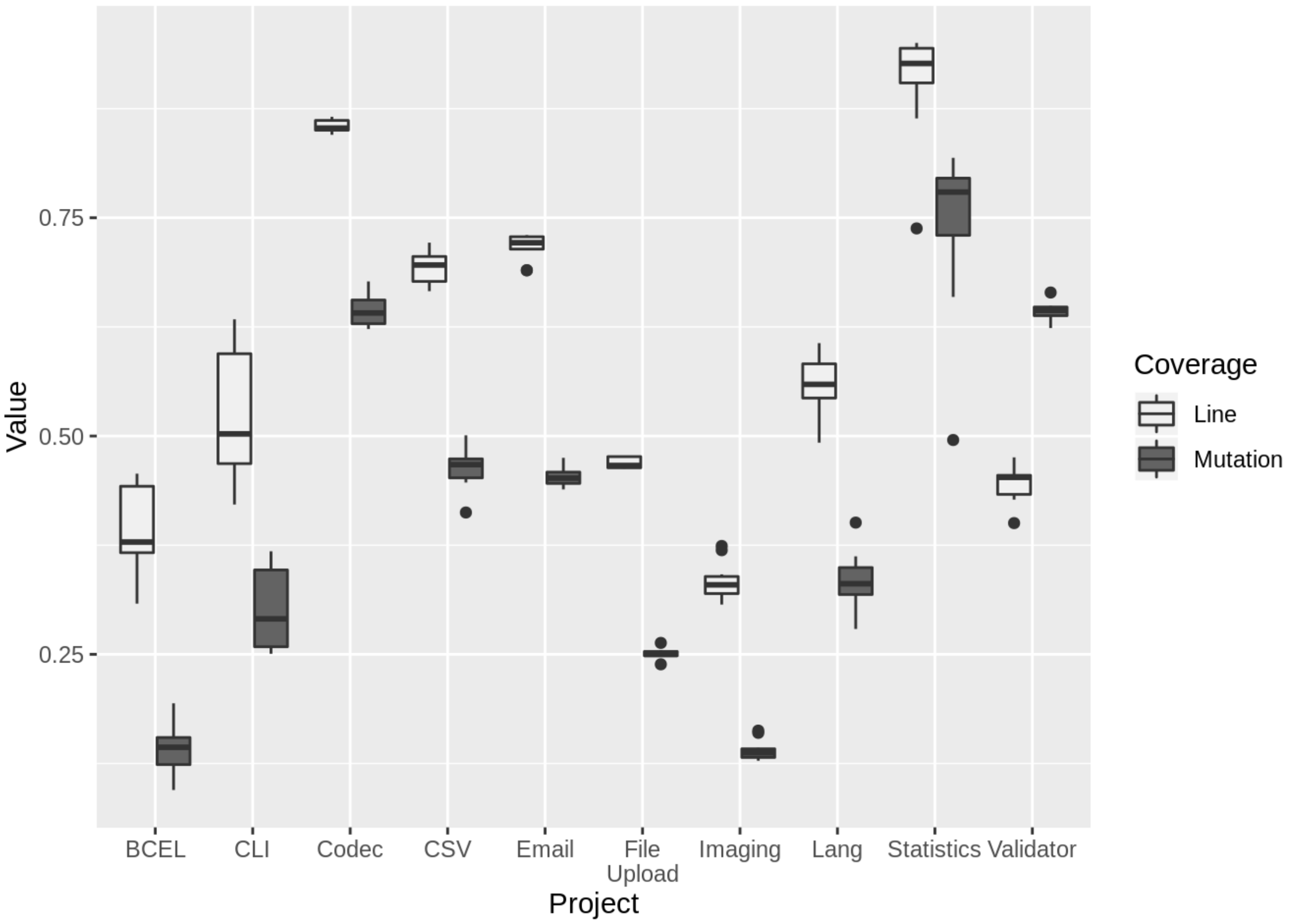
```
## # A tibble: 10 x 4
##    Project   LineCoverageAvera… LineCoverageVariat… LineCoverageStandardDe…
##    <fct>                  <dbl>               <dbl>                   <dbl>
##  1 bcel                   0.520            0.00147                   0.0383
##  2 cli                    0.772            0.00275                   0.0524
##  3 codec                  0.879            0.000141                  0.0119
##  4 csv                    0.344            0.00221                   0.0470
##  5 email                  0.384            0.00115                   0.0339
##  6 fileuplo…              0.681            0.00258                   0.0508
##  7 imaging                0.587            0.00769                   0.0877
##  8 lang                   0.446            0.00109                   0.0331
##  9 statisti…              0.938            0.00111                   0.0334
## 10 validator              0.639            0.00210                   0.0458
```

```
# Mutation Coverage mean, variation and standard deviation
evoMeanVariationSDMC <- evo %>% group_by(Project) %>% summarise(MutationCoverageAvera
ge = mean(MutationCoverage),
                                                               MutationCoverageVariati
on = var(MutationCoverage),
                                                               MutationCoverageStandar
dDeviation = sd(MutationCoverage))
evoMeanVariationSDMC
```

```
## # A tibble: 10 x 4
##    Project   MutationCoverageAv… MutationCoverageVa… MutationCoverageStand…
##    <fct>                   <dbl>              <dbl>                  <dbl>
##  1 bcel                    0.319           0.000983                 0.0314
##  2 cli                     0.510           0.0207                   0.144
##  3 codec                   0.696           0.00205                  0.0453
##  4 csv                     0.239           0.00423                  0.0650
##  5 email                   0.234           0.00148                  0.0385
##  6 fileuplo…               0.491           0.00299                  0.0547
##  7 imaging                 0.391           0.00547                  0.0740
##  8 lang                    0.318           0.000806                 0.0284
##  9 statisti…               0.724           0.000356                 0.0189
## 10 validator               0.536           0.00276                  0.0525
```

Here we analyse the difference between test suites generated for the same project with Randoop in terms of line and mutation coverage.

```
ggplot(data = randoopMelted, aes(x=Project, y=value)) +
  geom_boxplot(aes(fill=variable)) +
  scale_x_discrete(labels=projects) +
  labs(x ="Project", y = "Value") +
  scale_fill_manual(values = c("#f0f0f0", "#636363"), name = "Coverage",
                    labels = c("Line", "Mutation"))
```



```
# Line Coverage mean, variation and standard deviation
randoopMeanVariationSDLC <- randoop %>% group_by(Project) %>% summarise(LineCoverageA
verage = mean(LineCoverage),
                                                                 LineCoverageVariation =
var(LineCoverage),
                                                                 LineCoverageStandardDev
iation = sd(LineCoverage))
randoopMeanVariationSDLC
```

```
## # A tibble: 10 x 4
##    Project   LineCoverageAvera… LineCoverageVariat… LineCoverageStandardDe…
##    <fct>                  <dbl>               <dbl>                   <dbl>
##  1 bcel                   0.389           0.00294                    0.0543
##  2 cli                    0.524           0.00537                    0.0733
##  3 codec                  0.855           0.0000472                  0.00687
##  4 csv                    0.692           0.000353                   0.0188
##  5 email                  0.717           0.000233                   0.0153
##  6 fileuplo…              0.469           0.0000431                  0.00657
##  7 imaging                0.334           0.000488                   0.0221
##  8 lang                   0.559           0.00109                    0.0330
##  9 statisti…              0.905           0.00417                    0.0646
## 10 validator              0.447           0.000507                   0.0225
```

```
# Mutation Coverage mean, variation and standard deviation
randoopMeanVariationSDMC <- randoop %>% group_by(Project) %>% summarise(MutationCover
ageAverage = mean(MutationCoverage),
                                                              MutationCoverageVariati
on = var(MutationCoverage),
                                                              MutationCoverageStandar
dDeviation = sd(MutationCoverage))
randoopMeanVariationSDMC
```

```
## # A tibble: 10 x 4
##    Project   MutationCoverageAv… MutationCoverageVa… MutationCoverageStand…
##    <fct>                   <dbl>               <dbl>                  <dbl>
##  1 bcel                    0.142           0.000864                 0.0294
##  2 cli                     0.303           0.00226                  0.0475
##  3 codec                   0.643           0.000324                 0.0180
##  4 csv                     0.465           0.000650                 0.0255
##  5 email                   0.454           0.000123                 0.0111
##  6 fileuplo…               0.251           0.0000393                0.00627
##  7 imaging                 0.140           0.000143                 0.0119
##  8 lang                    0.334           0.00108                  0.0329
##  9 statisti…               0.741           0.00977                  0.0989
## 10 validator               0.643           0.000116                 0.0108
```

Here we compare the standard deviations of line and mutation coverage of the test suites generated with EvoSuite and Randoop.

```
# Line Coverage
wilcox.test(evoMeanVariationSDLC$LineCoverageStandardDeviation, randoopMeanVariationS
DLC$LineCoverageStandardDeviation, paired = TRUE)
```

```
## 
##  Wilcoxon signed rank test
## 
## data:  evoMeanVariationSDLC$LineCoverageStandardDeviation and randoopMeanVariation
SDLC$LineCoverageStandardDeviation
## V = 39, p-value = 0.2754
## alternative hypothesis: true location shift is not equal to 0
```

```
# Mutation Coverage
wilcox.test(evoMeanVariationSDMC$MutationCoverageStandardDeviation, randoopMeanVariat
ionSDMC$MutationCoverageStandardDeviation, paired = TRUE)
```

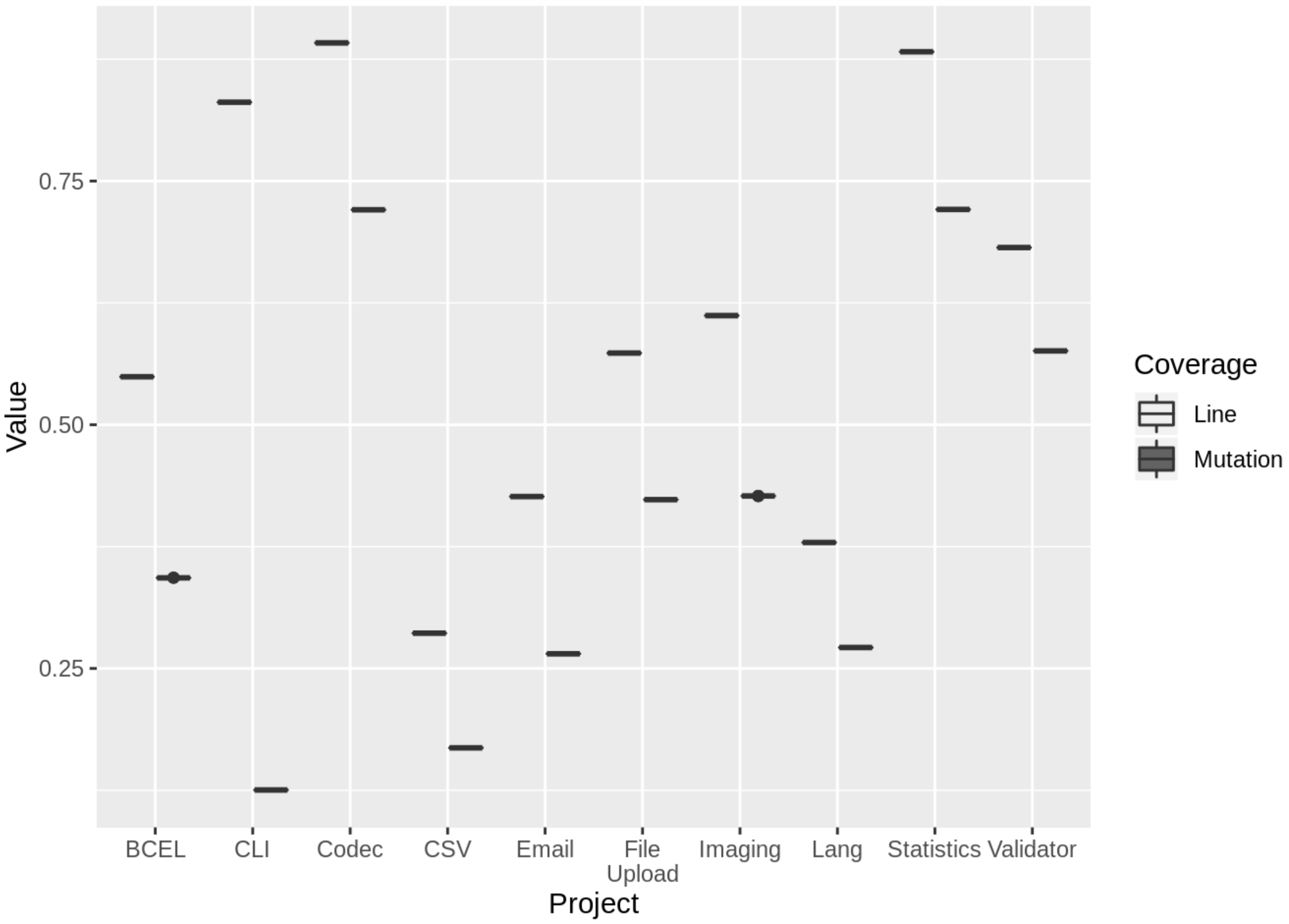
```
## 
##  Wilcoxon signed rank test
## 
## data:  evoMeanVariationSDMC$MutationCoverageStandardDeviation and randoopMeanVaria
tionSDMC$MutationCoverageStandardDeviation
## V = 44, p-value = 0.1055
## alternative hypothesis: true location shift is not equal to 0
```

PIT was executed 10 times for each test suite.

Here we analyse PIT’s outputs when executed for the same project and the same test suites

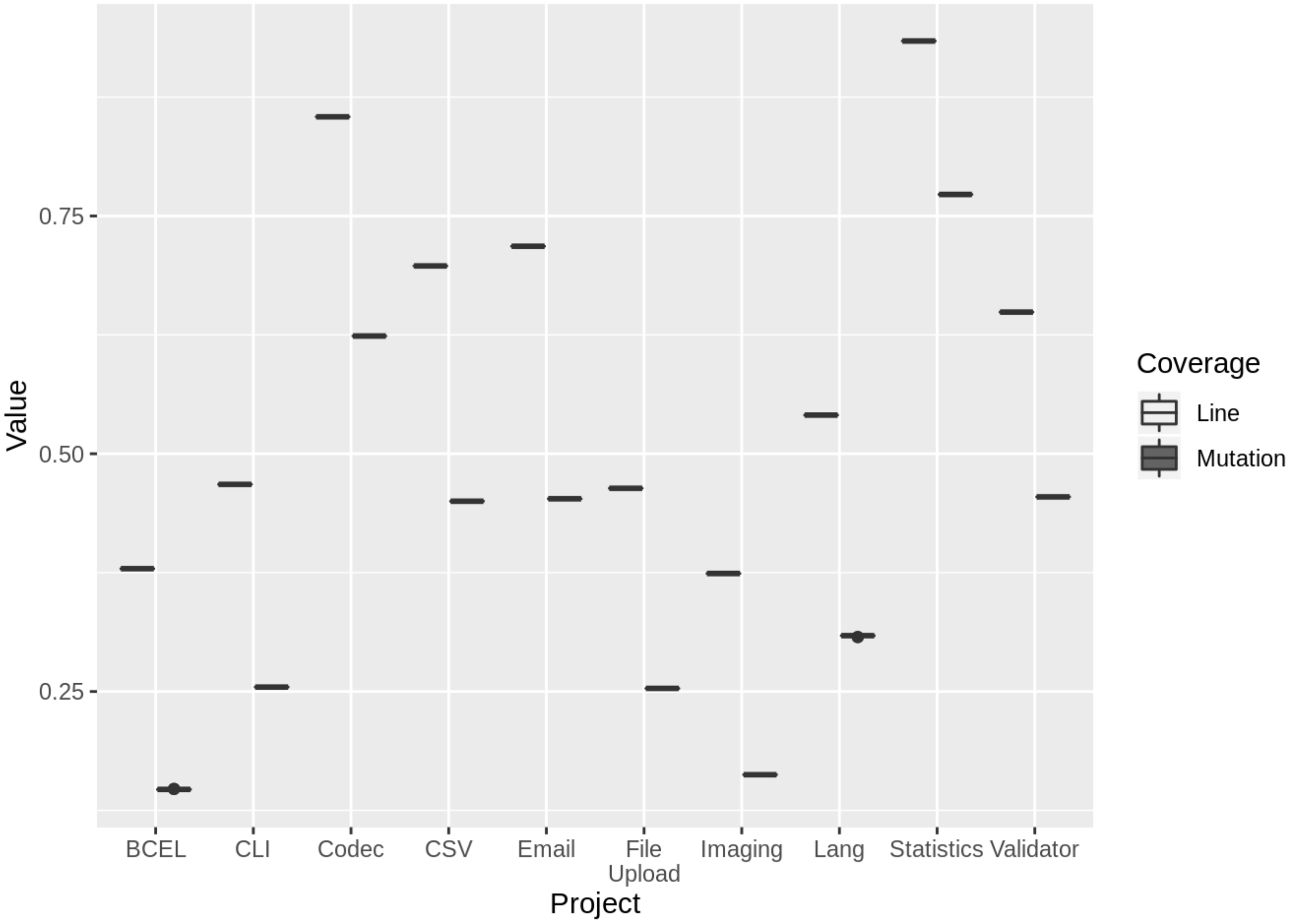
EvoSuite

```
ggplot(data = pitEvoMelted, aes(x=Project, y=value)) +
  geom_boxplot(aes(fill=variable)) +
  scale_x_discrete(labels=projects) +
  labs(x = "Project", y = "Value") +
  scale_fill_manual(values = c("#f0f0f0", "#636363"), name = "Coverage",
                    labels = c("Line", "Mutation"))
```



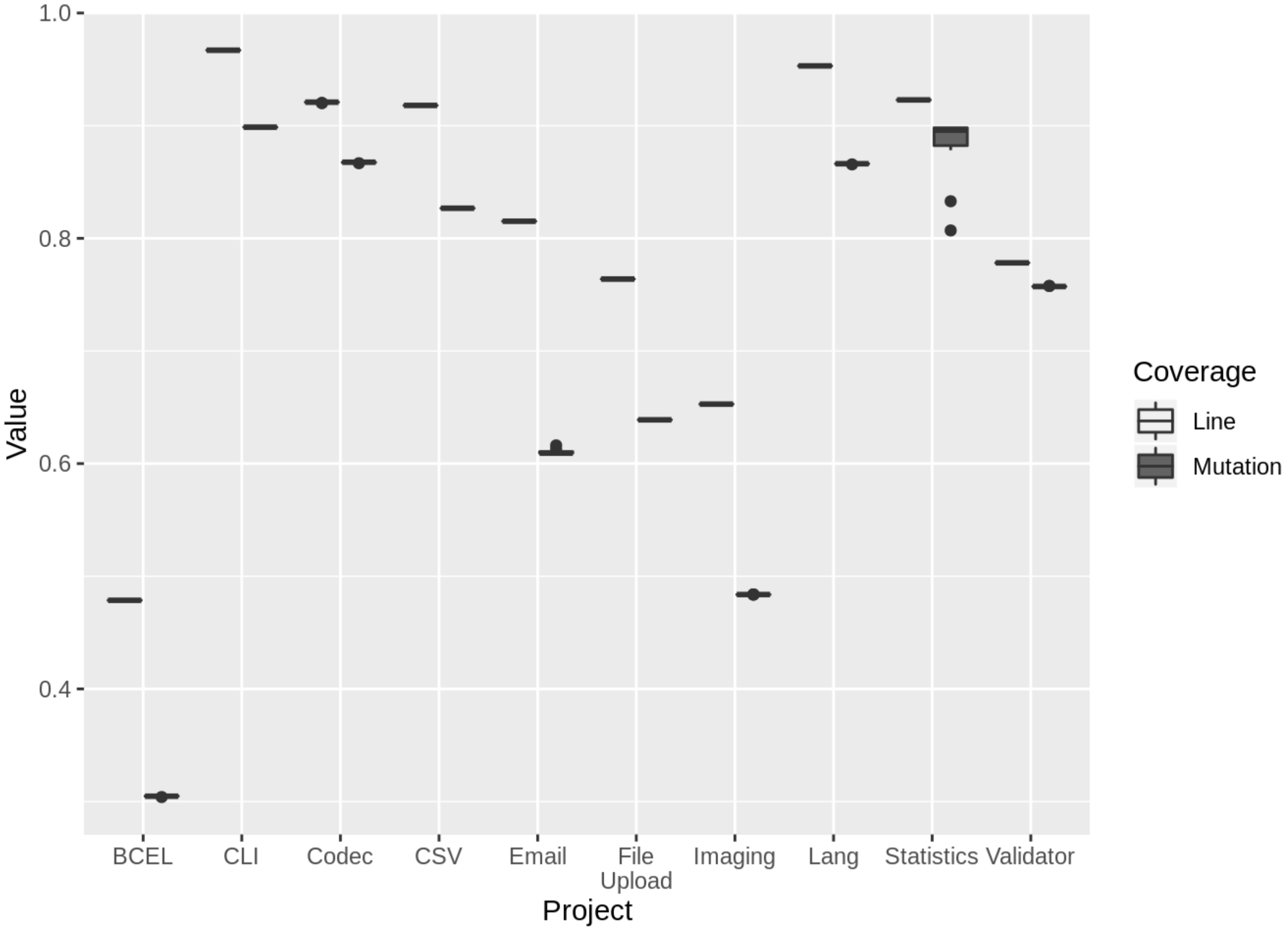
Randoop

```
ggplot(data = pitRandoopMelted, aes(x=Project, y=value)) +
  geom_boxplot(aes(fill=variable)) +
  scale_x_discrete(labels=projects) +
  labs(x ="Project", y = "Value") +
  scale_fill_manual(values = c("#f0f0f0", "#636363"), name = "Coverage",
                    labels = c("Line", "Mutation"))
```



Manual

```
ggplot(data = pitManualMelted, aes(x=Project, y=value)) +
  geom_boxplot(aes(fill=variable)) +
  scale_x_discrete(labels=projects) +
  labs(x ="Project", y = "Value") +
  scale_fill_manual(values = c("#f0f0f0", "#636363"), name = "Coverage",
                  labels = c("Line", "Mutation"))
```


1.0
0.8
0.6
0.4
Value
BCEL
CLI
Codec
CSV
Email
File Upload
Imaging
Lang
Statistics
Validator
Project
Coverage
Line
Mutation