Exploring Correlation Between The Amount of Times a Mutation Type Appears and Its Detection

Beatriz Souza

6/3/2019

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)
```

```
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
```

Reading data

```
#evoResults1 <- read.csv('/home/beatriz/Desktop/FinaldataSet/evoResultsEvoAvg.csv', s
ep = ",", header = TRUE)
#evoResults2 <- read.csv('/home/beatriz/Desktop/FinaldataSet/evoResults2EvoAvg.csv',
 sep = ",", header = TRUE)
#evoResults3 <- read.csv('/home/beatriz/Desktop/FinaldataSet/evoResults3EvoAvg.csv',
 sep = ",", header = TRUE)

#randoopResults1 <- read.csv('/home/beatriz/Desktop/FinaldataSet/randoopResultsRandoo
pAvg.csv', sep = ",", header = TRUE)
#randoopResults2 <- read.csv('/home/beatriz/Desktop/FinaldataSet/randoopResults2Rando
opAvg.csv', sep = ",", header = TRUE)
#randoopResults3 <- read.csv('/home/beatriz/Desktop/FinaldataSet/randoopResults3Rando
opAvg.csv', sep = ",", header = TRUE)

#manualResults1 <- read.csv('/home/beatriz/Desktop/FinaldataSet/manualSuitesResultsPI
TAvg.csv', sep = ",", header = TRUE)
#manualResults2 <- read.csv('/home/beatriz/Desktop/FinaldataSet/manualSuitesResults2P
ITAvg.csv', sep = ",", header = TRUE)
#manualResults3 <- read.csv('/home/beatriz/Desktop/FinaldataSet/manualSuitesResults3P
ITAvg.csv', sep = ",", header = TRUE)

mutantsGenerated <- read.csv('/home/beatriz/Documents/mutantsGenerated.csv', sep =
",", header = TRUE)
#mutantsGenerated

percentageOfGeneratedMutationsPerProject <- read.csv('/home/beatriz/Desktop/generated
MutantsPerProject.csv', sep = ",", header = TRUE)
percentageOfGeneratedMutationsPerProject
```

```
##        Project InvertNegsMutator ReturnValsMutator
## 1        BCEL      0.000000000         0.2136889
## 2         CLI      0.000000000         0.2960000
## 3       Codec      0.000000000         0.2226190
## 4         CSV      0.000000000         0.2864157
## 5       Email      0.000000000         0.3287435
## 6  FileUpload      0.000000000         0.2875817
## 7      Imaging      0.001036717         0.1686393
## 8        Lang      0.001532097         0.3009805
## 9  Statistics      0.022668394         0.2487047
## 10  Validator      0.000000000         0.3629113
##     ConditionalsBoundaryMutator VoidMethodCallMutator IncrementsMutator
## 1                    0.06623249            0.33484886       0.022855739
## 2                    0.04666667            0.15600000       0.010666667
## 3                    0.06755952            0.09136905       0.047321429
## 4                    0.04909984            0.10147299       0.019639935
## 5                    0.03614458            0.20137694       0.008605852
## 6                    0.06045752            0.13562092       0.035947712
## 7                    0.08924406            0.12164147       0.034902808
## 8                    0.10341658            0.07629845       0.033016700
## 9                    0.13147668            0.00000000       0.003238342
## 10                   0.11665005            0.05383848       0.016949153
##     NegateConditionalsMutator MathMutator
## 1                   0.3110101  0.05136397
## 2                   0.4546667  0.03600000
## 3                   0.2901786  0.28095238
## 4                   0.4615385  0.08183306
## 5                   0.4010327  0.02409639
## 6                   0.3643791  0.11601307
## 7                   0.2607343  0.32380130
## 8                   0.3621878  0.12256779
## 9                   0.1716321  0.42227979
## 10                  0.4062812  0.04336989
```

```
evoResults <- read.csv('/home/beatriz/Desktop/evoPercentageOfKM.csv', sep = ",", head
er = TRUE)
randoopResults <- read.csv('/home/beatriz/Desktop/randoopPercentageOfKM.csv', sep =
",", header = TRUE)
randoopResults
```

```
##      X MutationCoverage ReturnValsMutator ConditionalsBoundaryMutator
## 1    2        0.1407471         0.2441633                  0.11502783
## 2    4        0.3025333         0.5387387                  0.11142857
## 3    9        0.6430357         0.7030749                  0.49647577
## 4    8        0.4646481         0.5828571                  0.23333333
## 5    7        0.4535284         0.5732984                  0.23333333
## 6    3        0.2506536         0.3659091                  0.14324324
## 7    1        0.1402246         0.2627561                  0.06873185
## 8    5        0.3148940         0.3860130                  0.18798667
## 9   10        0.7407383         0.8575521                  0.31970443
## 10   6        0.4467597         0.5890110                  0.33803419
##      IncrementsMutator NegateConditionalsMutator MathMutator LineCoverage
## 1            0.1591398                 0.1658633   0.1167464    0.3889611
## 2            0.4125000                 0.2504399   0.1259259    0.5235149
## 3            0.5981132                 0.7218462   0.5826271    0.8550113
## 4            0.3750000                 0.5131206   0.0980000    0.6917301
## 5            0.7200000                 0.5206009   0.3285714    0.7168344
## 6            0.1590909                 0.2677130   0.2267606    0.4693772
## 7            0.1247525                 0.1556991   0.1175827    0.3336928
## 8            0.3703989                 0.3648787   0.2149812    0.5535309
## 9            0.7000000                 0.8641509   0.7492331    0.9050557
## 10           0.2911765                 0.4101840   0.3413793    0.6429462
##         Project    Tool InvertNegsMutator VoidMethodCallMutator
## 1          bcel randoop                NA            0.05893578
## 2           cli randoop                NA            0.09658120
## 3         codec randoop                NA            0.56384365
## 4           csv randoop                NA            0.33548387
## 5        e-mail randoop                NA            0.16752137
## 6    fileupload randoop                NA            0.05301205
## 7       imaging randoop         0.1500000            0.05426136
## 8          lang randoop         0.3900000            0.15237276
## 9    statistics randoop         0.8142857                    NA
## 10    validator randoop                NA            0.13333333
```

```
manualResults <- read.csv('/home/beatriz/Desktop/manualPercentageOfKM.csv', sep = ","
, header = TRUE)
manualResults
```

```
##      X MutationCoverage ReturnValsMutator ConditionalsBoundaryMutator
## 1   1        0.3048661         0.3513514                   0.3432282
## 2  10        0.8986667         0.8468468                   0.6571429
## 3   8        0.8672619         0.8302139                   0.6563877
## 4   6        0.8267091         0.8974359                   0.5333333
## 5   3        0.6161790         0.5863874                   0.2857143
## 6   4        0.6388889         0.5852273                   0.4324324
## 7   2        0.4836810         0.4124688                   0.4151261
## 8   7        0.8656351         0.8918300                   0.5274074
## 9   9        0.8797280         0.8216146                   0.7187192
## 10  5        0.7572283         0.7596154                   0.6153846
##     IncrementsMutator NegateConditionalsMutator MathMutator LineCoverage
## 1           0.4946237                 0.3615172   0.3803828    0.4786764
## 2           1.0000000                 0.9618768   0.9259259    0.9669967
## 3           0.9056604                 0.9353846   0.8781780    0.9208742
## 4           0.7272727                 0.8936170   0.3673469    0.9180000
## 5           1.0000000                 0.7467811   0.7142857    0.8151876
## 6           0.6363636                 0.7982063   0.7464789    0.7638408
## 7           0.7116705                 0.5370138   0.5001359    0.6528049
## 8           0.9535963                 0.9640440   0.8100000    0.9531415
## 9           1.0000000                 0.9671698   0.9259202    0.9228792
## 10          0.8529412                 0.8269939   0.8505747    0.7782152
##        Project   Tool InvertNegsMutator VoidMethodCallMutator
## 1         bcel manual                NA             0.1904587
## 2          cli manual                NA             0.8717949
## 3        codec manual                NA             0.8436482
## 4          csv manual                NA             0.8225806
## 5       e-mail manual                NA             0.4358974
## 6   fileupload manual                NA             0.3253012
## 7      imaging manual         0.1666667             0.4241540
## 8         lang manual         0.9500000             0.8032129
## 9   statistics manual         0.9114286                    NA
## 10   validator manual                NA             0.4166667
```

Merging data

```
#evoResults12 <- merge(evoResults1, evoResults2, all = TRUE)
#evo12
#evoResults123 <- merge(evoResults12, evoResults3, all = TRUE)
#evoResults123
#evoResults123 %>%  write.csv('/home/beatriz/Desktop/evoPercentageOfKM.csv')

#randoopResults12 <- merge(randoopResults1, randoopResults2, all = TRUE)
#randoopResults12
#randoopResults123 <- merge(randoopResults12, randoopResults3, all = TRUE)
#randoopResults123
#randoopResults123 %>%  write.csv('/home/beatriz/Desktop/randoopPercentageOfKM.csv')

#manualResults12 <- merge(manualResults1, manualResults2, all = TRUE)
#manualResults12
#manualResults123 <- merge(manualResults12, manualResults3, all = TRUE)
#manualResults123
#manualResults123 %>%  write.csv('/home/beatriz/Desktop/manualPercentageOfKM.csv')

#evoAndRandoopDataSet <- merge(evoResults123, randoopResults123, all = TRUE)
#completeDataSet <- merge(evoAndRandoopDataSet, manualResults123, all = TRUE)
#completeDataSet
```

Melting data

```
mutantsGenerated.m <- melt(mutantsGenerated, id.var = "Font")
mutantsGenerated.m
```

```
##                Font variable  value
## 1 Gerados por PIT      INM 0.0016
## 2 Gerados por PIT      RVM 0.2426
## 3 Gerados por PIT      CBM 0.0878
## 4 Gerados por PIT     VMCM 0.1403
## 5 Gerados por PIT       IM 0.0300
## 6 Gerados por PIT      NCM 0.3176
## 7 Gerados por PIT       MM 0.1802
```

```
ggplot(data=mutantsGenerated.m, aes(x=variable, y=value)) +
  geom_bar(stat="identity")
```
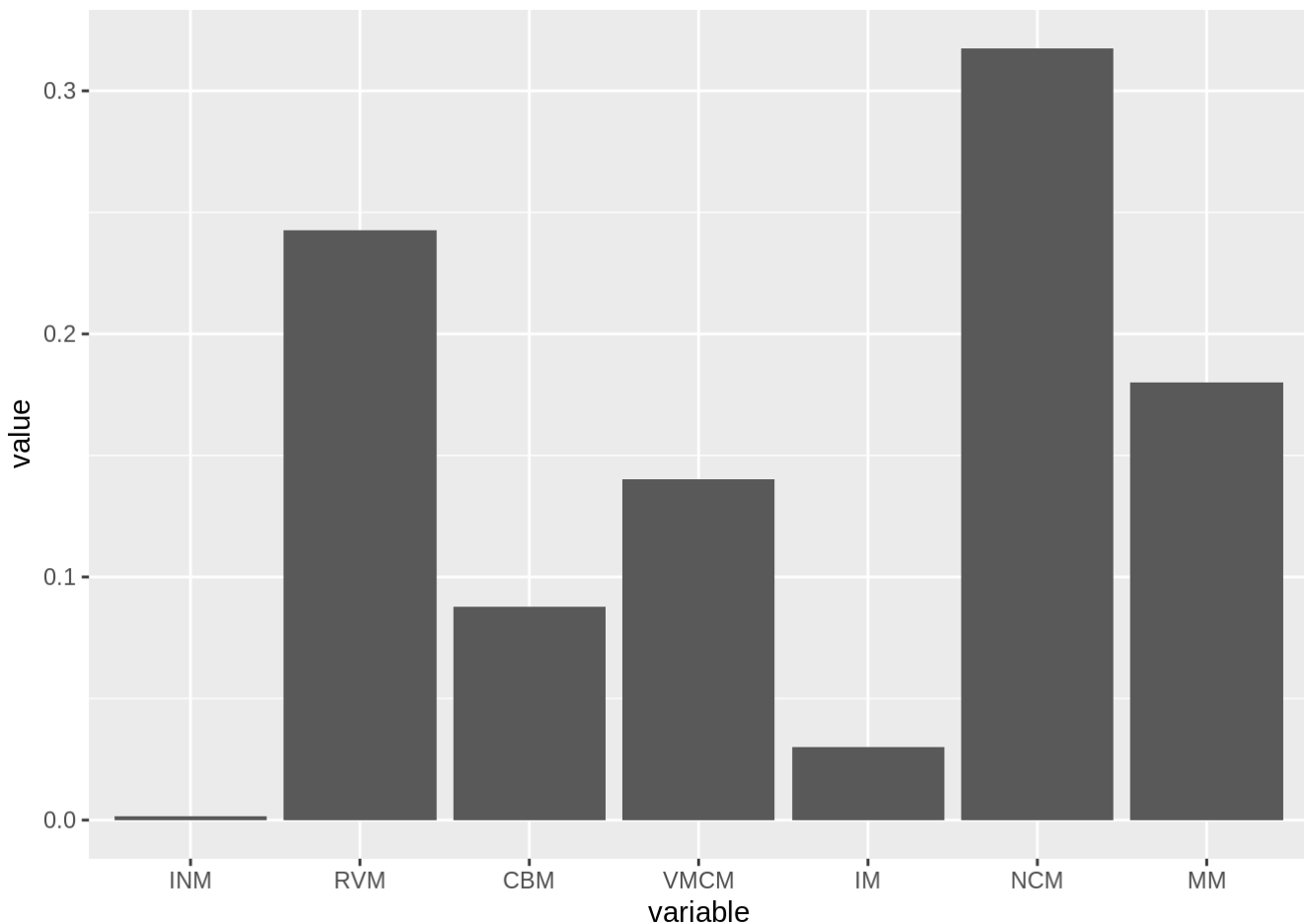
ReturnValsMutator

```
par(mfrow = c(2,2))
plot(evoResults$ReturnValsMutator ~ percentageOfGeneratedMutationsPerProject$ReturnVa
lsMutator)
plot(randoopResults$ReturnValsMutator ~ percentageOfGeneratedMutationsPerProject$Retu
rnValsMutator)
plot(manualResults$ReturnValsMutator ~ percentageOfGeneratedMutationsPerProject$Retur
nValsMutator)

cor.test(evoResults$ReturnValsMutator, percentageOfGeneratedMutationsPerProject$Retur
nValsMutator, method='spearman')
```
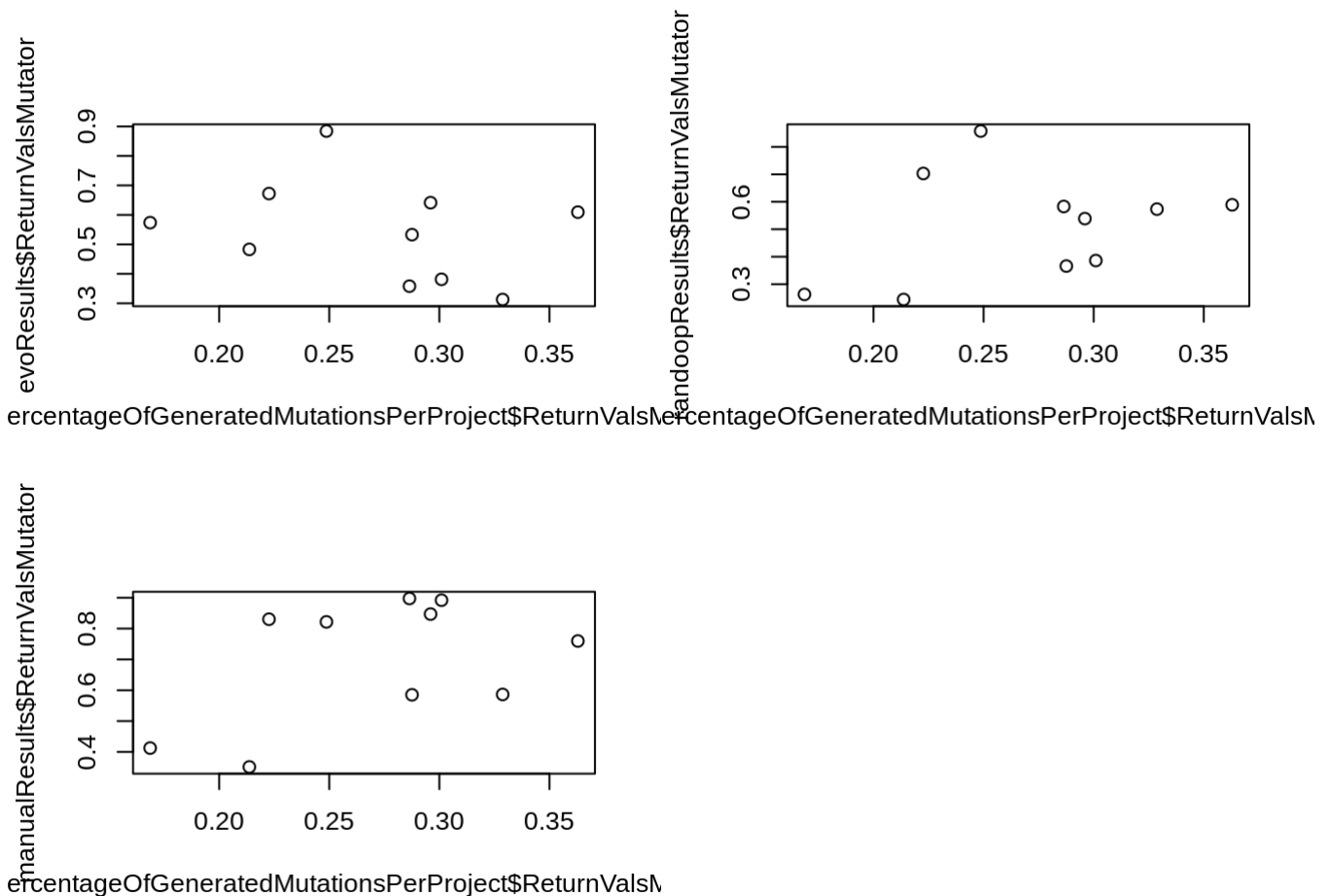
```
##
##  Spearman's rank correlation rho
##
## data:  evoResults$ReturnValsMutator and percentageOfGeneratedMutationsPerProject$R
eturnValsMutator
## S = 210, p-value = 0.4483
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.2727273
```

```
cor.test(randoopResults$ReturnValsMutator, percentageOfGeneratedMutationsPerProject$R
eturnValsMutator, method='spearman')
```

```
##
##  Spearman's rank correlation rho
##
## data:  randoopResults$ReturnValsMutator and percentageOfGeneratedMutationsPerProje
ct$ReturnValsMutator
## S = 120, p-value = 0.4483
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## 0.2727273
```

```
cor.test(manualResults$ReturnValsMutator, percentageOfGeneratedMutationsPerProject$Re
turnValsMutator, method='spearman')
```

```
##
##  Spearman's rank correlation rho
##
## data:  manualResults$ReturnValsMutator and percentageOfGeneratedMutationsPerProjec
t$ReturnValsMutator
## S = 108, p-value = 0.3305
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## 0.3454545
```

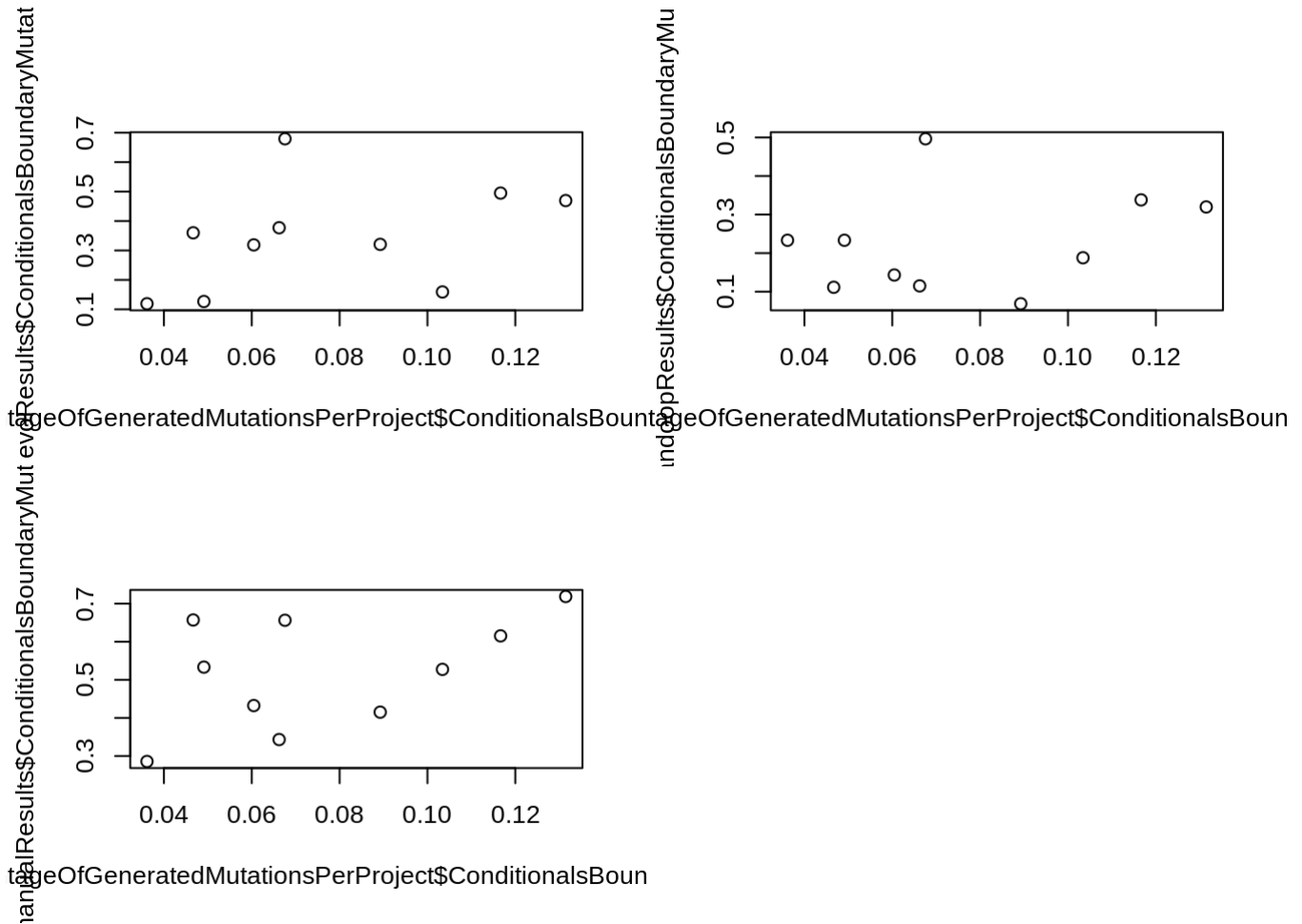# ConditionalsBoundaryMutator

```
par(mfrow = c(2,2))
plot(evoResults$ConditionalsBoundaryMutator ~ percentageOfGeneratedMutationsPerProjec
t$ConditionalsBoundaryMutator)
plot(randoopResults$ConditionalsBoundaryMutator ~ percentageOfGeneratedMutationsPerPr
oject$ConditionalsBoundaryMutator)
plot(manualResults$ConditionalsBoundaryMutator ~ percentageOfGeneratedMutationsPerPro
ject$ConditionalsBoundaryMutator)

cor.test(evoResults$ConditionalsBoundaryMutator, percentageOfGeneratedMutationsPerPro
ject$ConditionalsBoundaryMutator, method='spearman')
```

```
##
##  Spearman's rank correlation rho
##
## data:  evoResults$ConditionalsBoundaryMutator and percentageOfGeneratedMutationsPe
rProject$ConditionalsBoundaryMutator
## S = 70, p-value = 0.08777
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.5757576
```

```
cor.test(randoopResults$ConditionalsBoundaryMutator, percentageOfGeneratedMutationsPe
rProject$ConditionalsBoundaryMutator, method='spearman')
```

```
## Warning in cor.test.default(randoopResults$ConditionalsBoundaryMutator, :
## Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  randoopResults$ConditionalsBoundaryMutator and percentageOfGeneratedMutatio
nsPerProject$ConditionalsBoundaryMutator
## S = 111.84, p-value = 0.3639
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.3221899
```

```
cor.test(manualResults$ConditionalsBoundaryMutator, percentageOfGeneratedMutationsPer
Project$ConditionalsBoundaryMutator, method='spearman')
```

```
##
##  Spearman's rank correlation rho
##
## data:  manualResults$ConditionalsBoundaryMutator and percentageOfGeneratedMutation
sPerProject$ConditionalsBoundaryMutator
## S = 100, p-value = 0.2629
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.3939394
```

VoidMethodCallMutator

```
par(mfrow = c(2,2))
plot(evoResults$VoidMethodCallMutator ~ percentageOfGeneratedMutationsPerProject$Void
MethodCallMutator)
plot(randoopResults$VoidMethodCallMutator ~ percentageOfGeneratedMutationsPerProject
$VoidMethodCallMutator)
plot(manualResults$VoidMethodCallMutator ~ percentageOfGeneratedMutationsPerProject$V
oidMethodCallMutator)

cor.test(evoResults$VoidMethodCallMutator, percentageOfGeneratedMutationsPerProject$V
oidMethodCallMutator, method='spearman')
```

```
##
##  Spearman's rank correlation rho
##
## data:  evoResults$VoidMethodCallMutator and percentageOfGeneratedMutationsPerProje
ct$VoidMethodCallMutator
## S = 136, p-value = 0.7435
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.1333333
```

```
cor.test(randoopResults$VoidMethodCallMutator, percentageOfGeneratedMutationsPerProje
ct$VoidMethodCallMutator, method='spearman')
```

```
##
##  Spearman's rank correlation rho
##
## data:  randoopResults$VoidMethodCallMutator and percentageOfGeneratedMutationsPerP
roject$VoidMethodCallMutator
## S = 164, p-value = 0.3363
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.3666667
```

```
cor.test(manualResults$VoidMethodCallMutator, percentageOfGeneratedMutationsPerProjec
t$VoidMethodCallMutator, method='spearman')
```

```
##
##  Spearman's rank correlation rho
##
## data:  manualResults$VoidMethodCallMutator and percentageOfGeneratedMutationsPerPr
oject$VoidMethodCallMutator
## S = 148, p-value = 0.5517
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.2333333
```
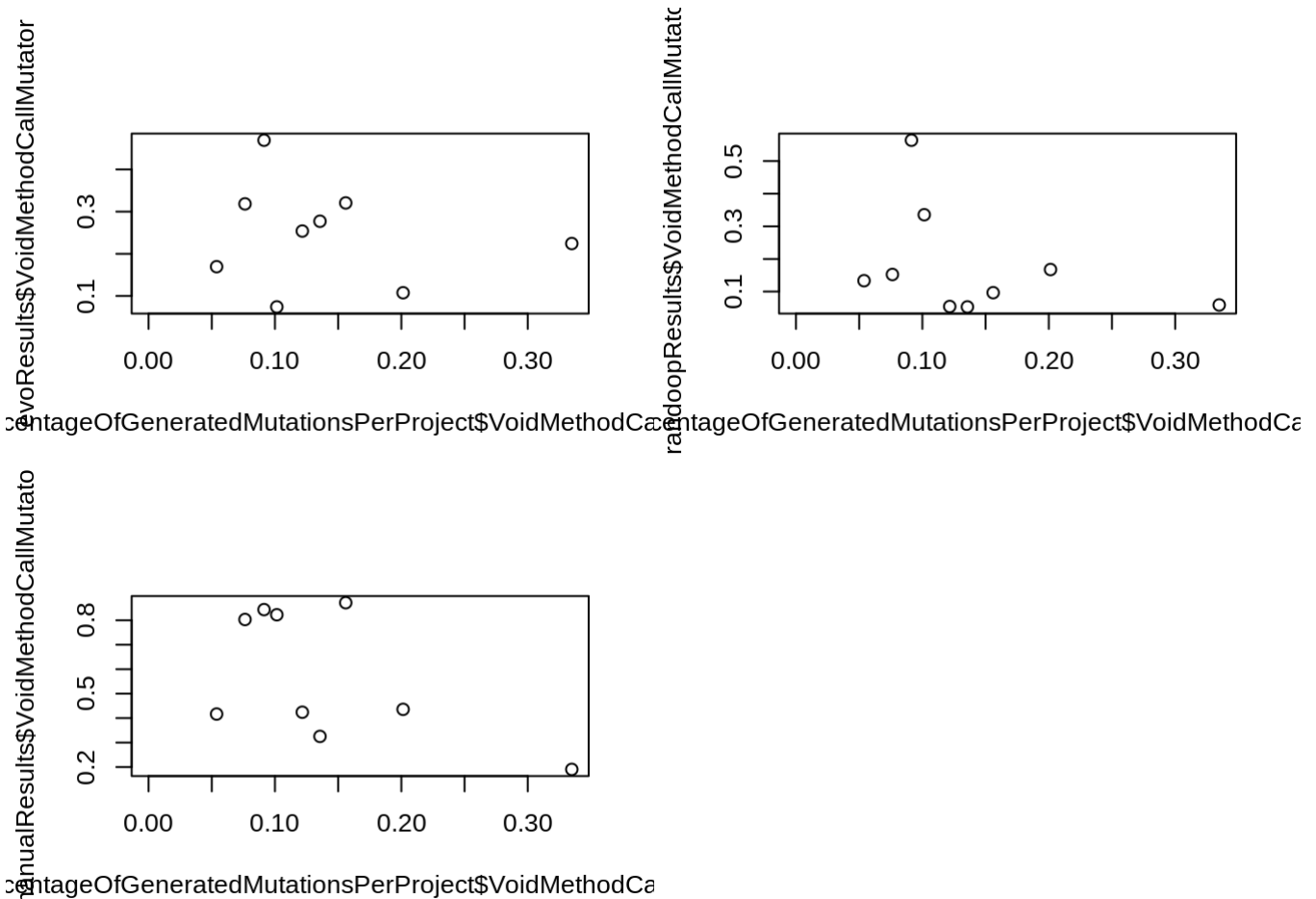

IncrementsMutator

```
par(mfrow = c(2,2))
plot(evoResults$IncrementsMutator ~ percentageOfGeneratedMutationsPerProject$Incremen
tsMutator)
plot(randoopResults$IncrementsMutator ~ percentageOfGeneratedMutationsPerProject$Incr
ementsMutator)
plot(manualResults$IncrementsMutator ~ percentageOfGeneratedMutationsPerProject$Incre
mentsMutator)

cor.test(evoResults$IncrementsMutator, percentageOfGeneratedMutationsPerProject$Incre
mentsMutator, method='spearman')
```

```
##
##  Spearman's rank correlation rho
##
## data:  evoResults$IncrementsMutator and percentageOfGeneratedMutationsPerProject$I
ncrementsMutator
## S = 146, p-value = 0.7588
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.1151515
```
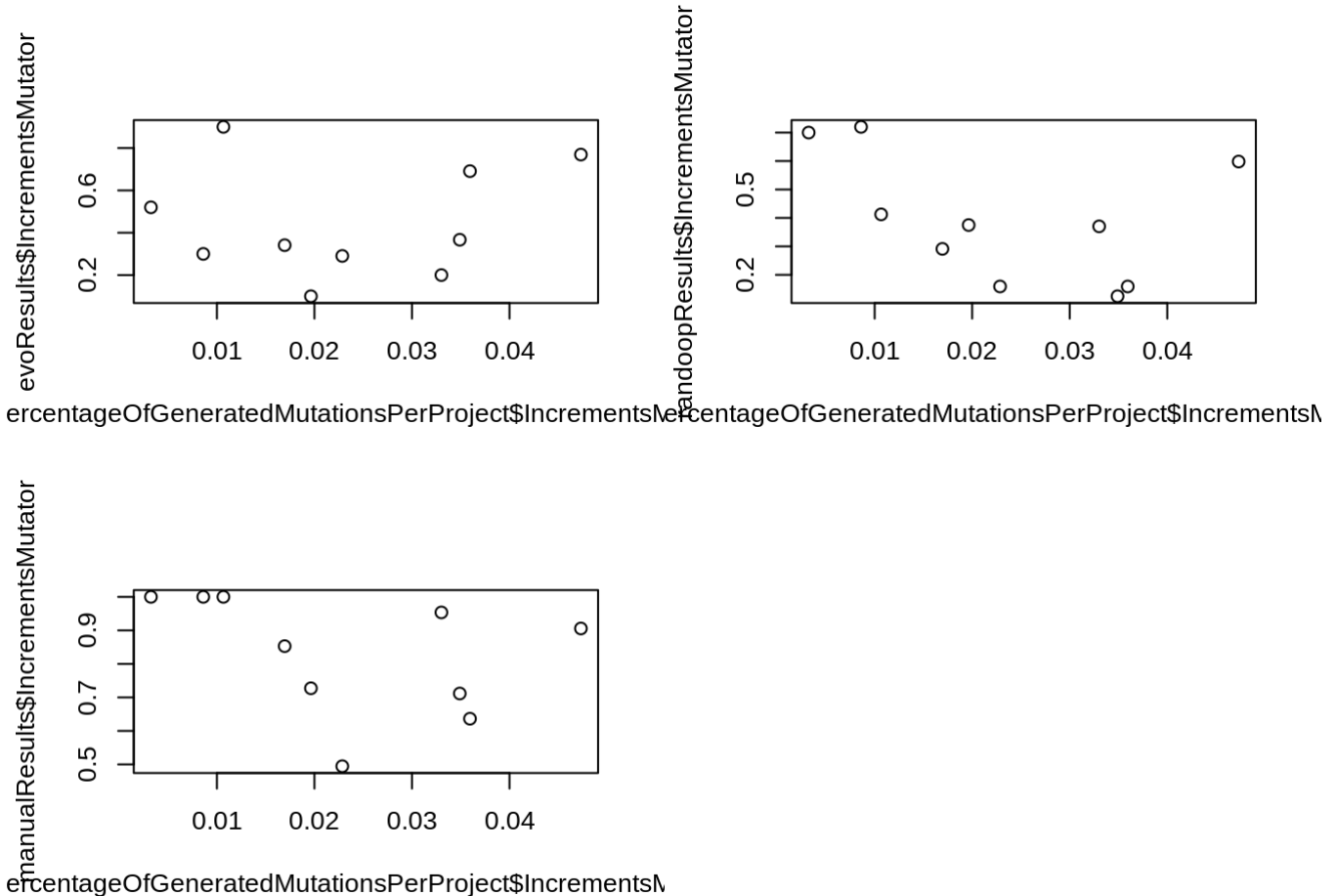
```
cor.test(randoopResults$IncrementsMutator, percentageOfGeneratedMutationsPerProject$I
ncrementsMutator, method='spearman')
```

```
##
##  Spearman's rank correlation rho
##
## data:  randoopResults$IncrementsMutator and percentageOfGeneratedMutationsPerProje
ct$IncrementsMutator
## S = 260, p-value = 0.08777
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.5757576
```

```
cor.test(manualResults$IncrementsMutator, percentageOfGeneratedMutationsPerProject$In
crementsMutator, method='spearman')
```

```
## Warning in cor.test.default(manualResults$IncrementsMutator,
## percentageOfGeneratedMutationsPerProject$IncrementsMutator, : Cannot
## compute exact p-value with ties
```

```
##
##   Spearman's rank correlation rho
##
## data:  manualResults$IncrementsMutator and percentageOfGeneratedMutationsPerProjec
t$IncrementsMutator
## S = 269.27, p-value = 0.04998
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##         rho
## -0.6319494
```


NegateConditionalsMutator

```
par(mfrow = c(2,2))
plot(evoResults$NegateConditionalsMutator ~ percentageOfGeneratedMutationsPerProject
$NegateConditionalsMutator)
plot(randoopResults$NegateConditionalsMutator ~ percentageOfGeneratedMutationsPerProj
ect$NegateConditionalsMutator)
plot(manualResults$NegateConditionalsMutator ~ percentageOfGeneratedMutationsPerProje
ct$NegateConditionalsMutator)

cor.test(evoResults$IncrementsMutator, percentageOfGeneratedMutationsPerProject$Negat
eConditionalsMutator, method='spearman')
```

```
##
##  Spearman's rank correlation rho
##
## data:  evoResults$IncrementsMutator and percentageOfGeneratedMutationsPerProject$N
egateConditionalsMutator
## S = 202, p-value = 0.5367
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.2242424
```
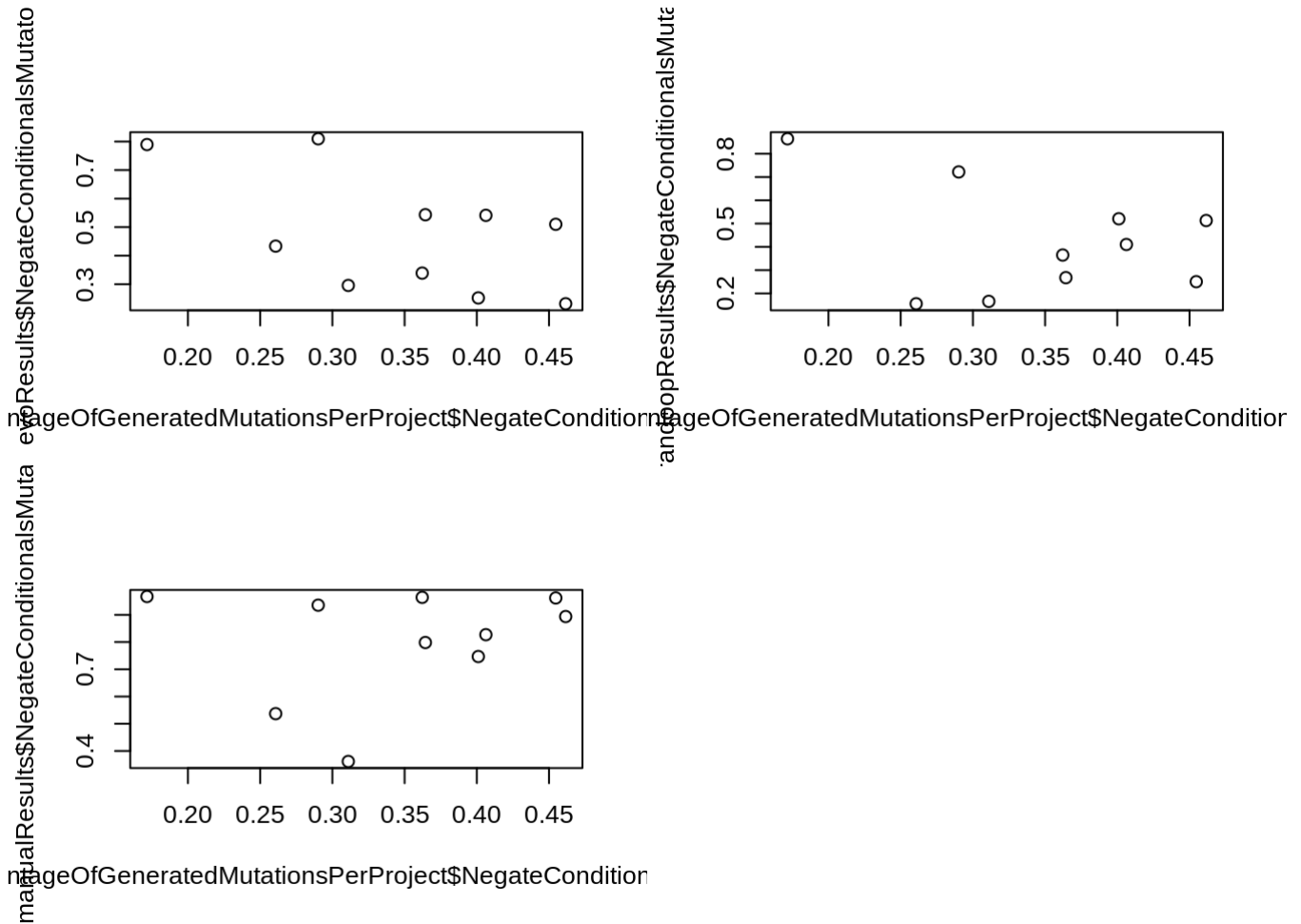
```
cor.test(randoopResults$IncrementsMutator, percentageOfGeneratedMutationsPerProject$N
egateConditionalsMutator, method='spearman')
```

```
##
##  Spearman's rank correlation rho
##
## data:  randoopResults$IncrementsMutator and percentageOfGeneratedMutationsPerProje
ct$NegateConditionalsMutator
## S = 152, p-value = 0.838
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## 0.07878788
```

```
cor.test(manualResults$IncrementsMutator, percentageOfGeneratedMutationsPerProject$Ne
gateConditionalsMutator, method='spearman')
```

```
## Warning in cor.test.default(manualResults$IncrementsMutator,
## percentageOfGeneratedMutationsPerProject$NegateConditionalsMutator, :
## Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  manualResults$IncrementsMutator and percentageOfGeneratedMutationsPerProjec
t$NegateConditionalsMutator
## S = 153.86, p-value = 0.853
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## 0.06748974
```

MathMutator

```
par(mfrow = c(2,2))
plot(evoResults$MathMutator ~ percentageOfGeneratedMutationsPerProject$MathMutator)
plot(randoopResults$MathMutator ~ percentageOfGeneratedMutationsPerProject$MathMutato
r)
plot(manualResults$MathMutator ~ percentageOfGeneratedMutationsPerProject$MathMutato
r)

cor.test(evoResults$IncrementsMutator, percentageOfGeneratedMutationsPerProject$MathM
utator, method='spearman')
```

```
##
##  Spearman's rank correlation rho
##
## data:  evoResults$IncrementsMutator and percentageOfGeneratedMutationsPerProject$M
athMutator
## S = 142, p-value = 0.7072
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## 0.1393939
```
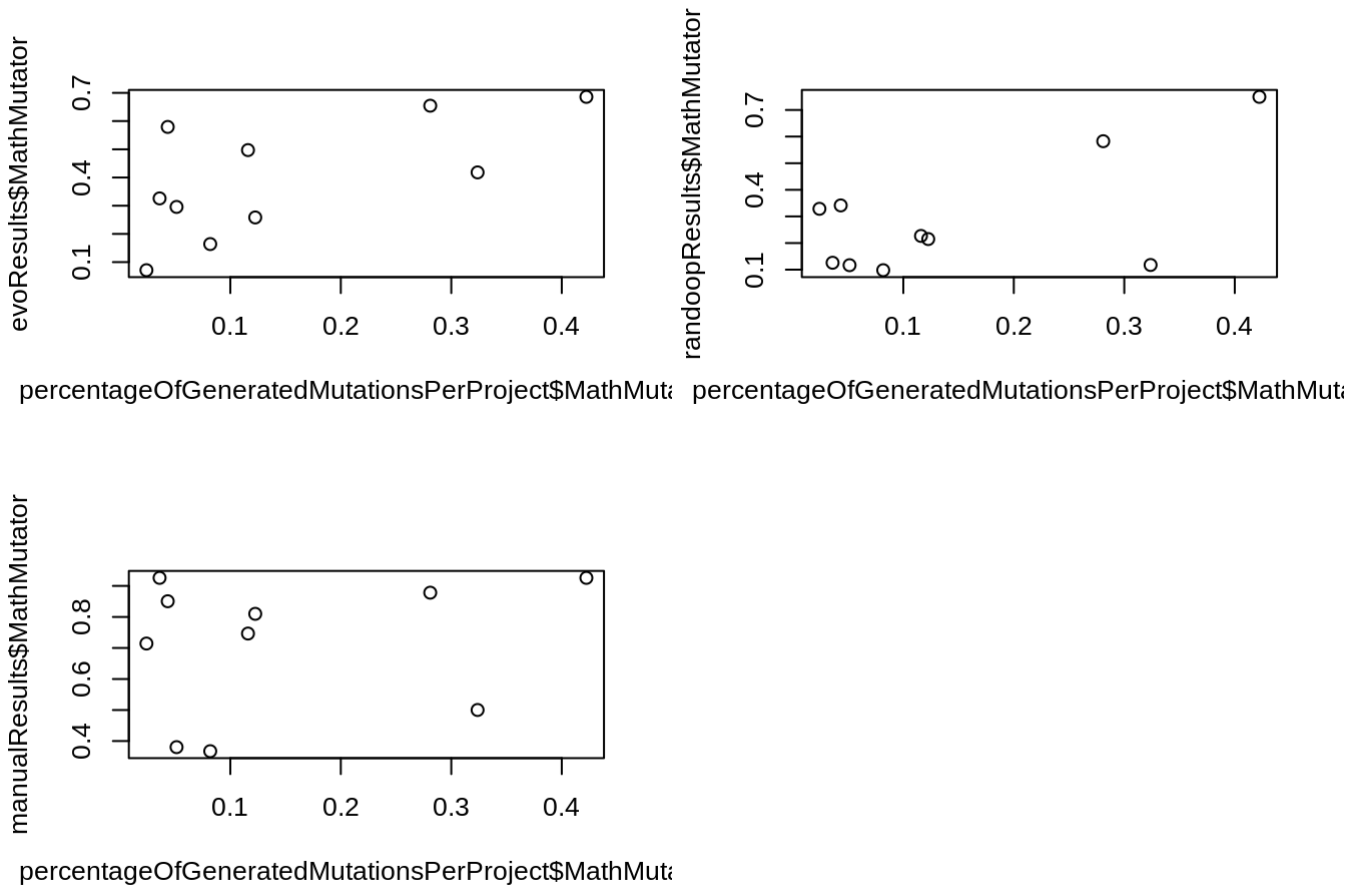
```
cor.test(randoopResults$IncrementsMutator, percentageOfGeneratedMutationsPerProject$M
athMutator, method='spearman')
```

```
##
##   Spearman's rank correlation rho
##
## data:  randoopResults$IncrementsMutator and percentageOfGeneratedMutationsPerProje
ct$MathMutator
## S = 194, p-value = 0.632
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##         rho
## -0.1757576
```

```
cor.test(manualResults$IncrementsMutator, percentageOfGeneratedMutationsPerProject$Ma
thMutator, method='spearman')
```

```
## Warning in cor.test.default(manualResults$IncrementsMutator,
## percentageOfGeneratedMutationsPerProject$MathMutator, : Cannot compute
## exact p-value with ties
```

```
##
##   Spearman's rank correlation rho
##
## data:  manualResults$IncrementsMutator and percentageOfGeneratedMutationsPerProjec
t$MathMutator
## S = 186.26, p-value = 0.7228
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##         rho
## -0.1288441
```

InvertNegsMutator

```
par(mfrow = c(2,2))
plot(evoResults$InvertNegsMutator ~ percentageOfGeneratedMutationsPerProject$InvertNe
gsMutator)
plot(randoopResults$InvertNegsMutator ~ percentageOfGeneratedMutationsPerProject$Inve
rtNegsMutator)
plot(manualResults$InvertNegsMutator ~ percentageOfGeneratedMutationsPerProject$Inver
tNegsMutator)

cor.test(evoResults$InvertNegsMutator, percentageOfGeneratedMutationsPerProject$Inver
tNegsMutator, method='spearman')
```

```
##
##  Spearman's rank correlation rho
##
## data:  evoResults$InvertNegsMutator and percentageOfGeneratedMutationsPerProject$I
nvertNegsMutator
## S = 0, p-value = 0.3333
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
## rho
##   1
```

```
cor.test(randoopResults$InvertNegsMutator, percentageOfGeneratedMutationsPerProject$I
nvertNegsMutator, method='spearman')
```

```
##
##  Spearman's rank correlation rho
##
## data:  randoopResults$InvertNegsMutator and percentageOfGeneratedMutationsPerProje
ct$InvertNegsMutator
## S = 0, p-value = 0.3333
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
## rho
##   1
```

```
cor.test(manualResults$InvertNegsMutator, percentageOfGeneratedMutationsPerProject$In
vertNegsMutator, method='spearman')
```

```
##
##  Spearman's rank correlation rho
##
## data:  manualResults$InvertNegsMutator and percentageOfGeneratedMutationsPerProjec
t$InvertNegsMutator
## S = 2, p-value = 1
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
## rho
## 0.5
```