

Calculation of processing and overprocessing

1 For Bondora log

```
print(getwd())

## [1] "/home/coderus/ownCloud/Education/PhD/Papers/caise2016overprocessing/source/output"

filenames <- list.files()[grep(paste("^output_bondora(=?.*\\_under.csv)", sep=''), list.files(), perl=TRUE)]
dat = c()
result = matrix(0,nrow = length(filenames),ncol = 6)
colnames(result) = c("nr_checks_our", "nr_checks_Wil", "nr_checks_rand",
                    "overproc_our", "overproc_Wil", "overproc_rand")

for (i in 1:length(filenames)) {
  foo = read.table(filenames[i], sep=",", header=TRUE)
  dat = rbind(dat,foo)
  result[i,1] = mean(foo$nr_checks_our)
  result[i,2] = mean(foo$nr_checks_Wil)
  result[i,3] = mean(foo$nr_checks_rand)
  result[i,4] = mean(foo$nr_checks_our - foo$minimum_check_number)
  result[i,5] = mean(foo$nr_checks_Wil - foo$minimum_check_number)
  result[i,6] = mean(foo$nr_checks_rand - foo$minimum_check_number)
}
```

Average number of checks that one would do if they follow **our ordering**:

```
round(mean(result[,1]), digits = 4)

## [1] 2.7937
```

Average number of checks that one would do if they apply **Wil's method** (constant reject probabilities):

```
round(mean(result[,2]), digits = 4)

## [1] 2.797
```

Average number of checks that one would do if for every case they do checks in **random order**

```
round(mean(result[,3]), digits = 4)

## [1] 2.8443
```

Average **overprocessing** - our method

```
round(mean(result[,4]), digits = 4)

## [1] 0.1009
```

Average **overprocessing** - Wil method

```
round(mean(result[,5]),digits = 4)

## [1] 0.1042
```

Average **overprocessing** - random ordering

```
round(mean(result[,6]),digits = 4)

## [1] 0.1514
```

```
tt = matrix(0,nrow = 3,ncol = length(unique(dat$nr_checks_rand)))
rownames(tt) = c("count_checks_our","count_checks_Wil","count_checks_rand")
tt[1,] = round(table(dat$nr_checks_our)/length(filenamees))
tt[2,] = round(table(dat$nr_checks_Wil)/length(filenamees))
tt[3,] = round(table(dat$nr_checks_rand)/length(filenamees))
colnames(tt) = names(table(dat$nr_checks_rand))
print(tt)

##              1    2    3
## count_checks_our 443 768 6802
## count_checks_Wil 637 352 7023
## count_checks_rand 428 392 7192
```

Distribution of overprocessing

2 For Environmental permit log

```
filenames <- list.files()[grep(paste("^output_envpermit(?:=.*\\)_under.csv",sep=' '), list.files()), perl=TRUE]
dat = c()
result = matrix(0,nrow = length(filenames),ncol = 6)
colnames(result) = c("nr_checks_our","nr_checks_Wil","nr_checks_rand",
                    "overproc_our","overproc_Wil","overproc_rand")

for (i in 1:length(filenames)) {
  foo = read.table(filenames[i],sep=",",header=TRUE)
  dat = rbind(dat,foo)
  result[i,1] = mean(foo$nr_checks_our)
  result[i,2] = mean(foo$nr_checks_Wil)
  result[i,3] = mean(foo$nr_checks_rand)
  result[i,4] = mean(foo$nr_checks_our - foo$minimum_check_number)
  result[i,5] = mean(foo$nr_checks_Wil - foo$minimum_check_number)
  result[i,6] = mean(foo$nr_checks_rand - foo$minimum_check_number)
}
```

Average number of checks that one would do if they follow **our ordering**:

```
round(mean(result[,1]),digits = 4)

## [1] 2.3337
```

Average number of checks that one would do if they apply **Wil's method** (constant reject probabilities):

```
round(mean(result[,2]),digits = 4)

## [1] 2.3309
```

Average number of checks that one would do if for every case they do checks in **random order**

```
round(mean(result[,3]),digits = 4)

## [1] 2.6626
```

Average **overprocessing** - our method

```
round(mean(result[,4]),digits = 4)

## [1] 0.6581
```

Average **overprocessing** - Wil method

```
round(mean(result[,5]),digits = 4)

## [1] 0.6553
```

Average **overprocessing** - random ordering

```
round(mean(result[,6]),digits = 4)

## [1] 0.987
```

```
tt = matrix(0,nrow = 3,ncol = length(unique(dat$nr_checks_rand)))
rownames(tt) = c("count_checks_our","count_checks_Wil","count_checks_rand")
tt[1,] = round(table(dat$nr_checks_our)/length(filenames))
tt[2,] = round(table(dat$nr_checks_Wil)/length(filenames))
tt[3,] = round(table(dat$nr_checks_rand)/length(filenames))
colnames(tt) = names(table(dat$nr_checks_rand))
print(tt)

##           1    2    3
## count_checks_our  3 158  85
## count_checks_Wil  3 159  84
## count_checks_rand 2   80 165
```

Distribution of overprocessing