

# Contrôle du kFWER à la Romano et Wolf

*Pierre Neuvial*

## ENSAE ParisTech: TP tests multiples

Romano et Wolf ont proposé en 2007 une procédure reposant sur des ré-échantillonnages, qui offre un contrôle asymptotique du k-FWER. Cette procédure est décrite dans un article de survey de 2008, voir références ci-dessous.

### Procédure ‘generalized StepM’

- Comprendre la généralisation de la procédure “StepM” décrite en section 4.3 de [1]. Evaluer sa complexité algorithmique (en nombre d’opérations).
- Expliquer le “raccourci” proposé en remarque 4.1, et donner sa complexité algorithmique.

### Application

Ce raccourci est implémenté sur le site de M. Wolf: <http://www.econ.uzh.ch/faculty/wolf/publications.html>  
J’ai créé un package R (minimal) à partir des scripts disponibles.

- L’installer depuis R:

```
install.packages("http://www.math-evry.cnrs.fr/_media/members/pneuvial/ensae/tp/kfwe_0.0.1.tar.gz")
```

Si cela ne fonctionne pas, voir Annexe.

- Appliquer la fonction ‘k.fwe’ sur les données d’exemple fournies dans le package.

```
library("kFWE")
data("Z2,RSW08", package="kFWE")
data("Z02,RSW08", package="kFWE")

system.time(resRSW3 <- k.fwe(dat, dat0, k=3, alpha=0.05))
```

```
## user system elapsed
## 182.282 1.203 183.611
```

### Représentation graphique des résultats

```
rg <- range(c(dat, dat0))
cols <- rep(1, length(dat))
pchs <- rep(1, length(dat))
cols[resRSW3$hypo.rej] <- cols[resRSW3$hypo.rej] + resRSW3$step.rej
pchs[resRSW3$hypo.rej] <- 19
```

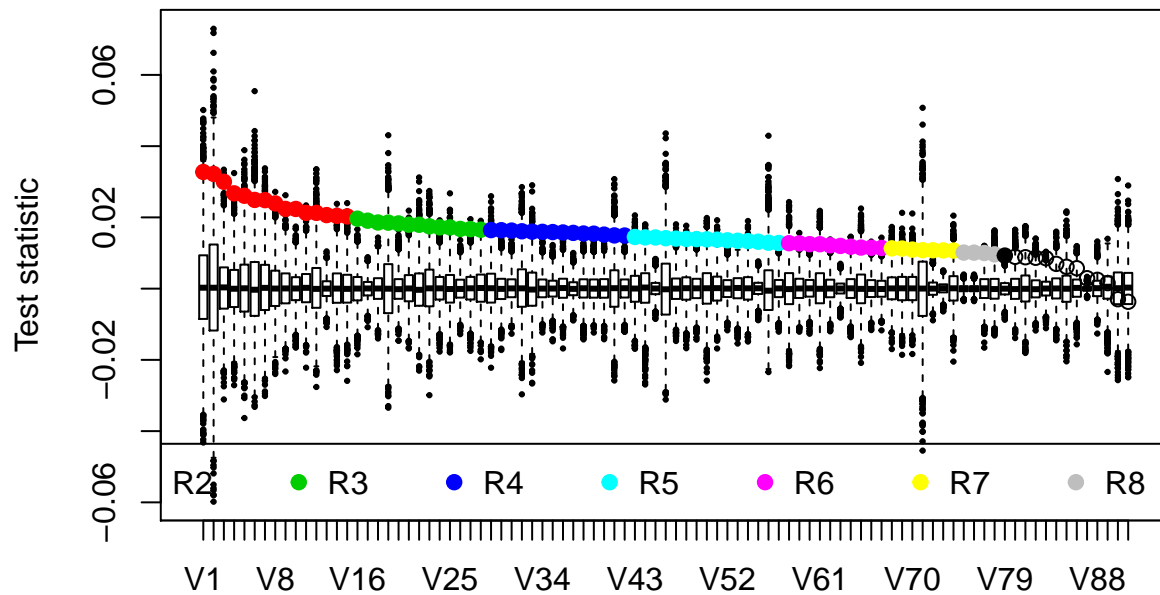
```

maxrej <- max(resRSW3$step.rej)
lgd <- c( paste("R", 1:maxrej, sep=""), "not rej.")

boxplot(dat0, ylim=rg, pch=19, cex=0.3,
        main="Resampling-based kFWER control (RSW08)",
        xlab="", ylab="Test statistic")
points(dat, col=cols, pch=pchs)
legend("bottom", lgd, col=c(1:maxrej, 1), pch=c(rep(19, maxrej), 1), horiz=TRUE)

```

## Resampling-based kFWER control (RSW08)



- Comparer avec deux autres méthodes de contrôle du kFWER: la méthode de Bonferroni généralisée, et avec la méthode de Bonferroni-Holm généralisée.

## Generalized Holm

```

genHolm <- function(p, alpha, k, doPlot=FALSE) {
  S <- length(p)
  j <- 1:S
  jk <- (sign(j-k)==1)*(j-k) ## positive part of j-k
  tau <- alpha*k/(S-jk)
  rk <- rank(p)
  o <- order(p)
  po <- p[o]
  isUnder <- (po < tau)
  j0 <- min(which(cumsum(isUnder)<j))-1
  if (j0==0) {
    res <- numeric(0)
  } else {
    res <- 1:j0
  }
}

```

```

if (doPlot) {
  cols <- c(rep(2, j0), rep(1, S-j0))
  plot(po, col=cols, ylim=c(0, alpha))
  lines(tau, cex=0.5, lty=1)
}
## back to original indices
y <- sort(o[res])
attr(y, 'tau') <- tau
y
}

```

## Generalized Bonferroni

```

genBonf <- function(p, alpha, k) {
  S <- length(p)
  thr <- alpha*k/S
  y <- which(p<thr)
  attr(y, 'tau') <- rep(thr, S)
  y
}

```

## p-values

```

pval <- sapply(1:length(dat), FUN=function(ii) {
  stat <- dat[ii]
  stat0 <- dat0[, ii]
  B <- length(stat0)
  sum(stat<stat0)/(B+1)
})
alpha <- 0.05
resHolm <- genHolm(pval, alpha=alpha, k=3)
resBonf <- genBonf(pval, alpha=alpha, k=3)

cols <- rep(1, length(dat))
cols[resBonf] <- 1
cols[setdiff(resHolm, resBonf)] <- 2

pchs <- rep(1, length(dat))
pchs[resHolm] <- 19

```

## Représentation graphique des résultats

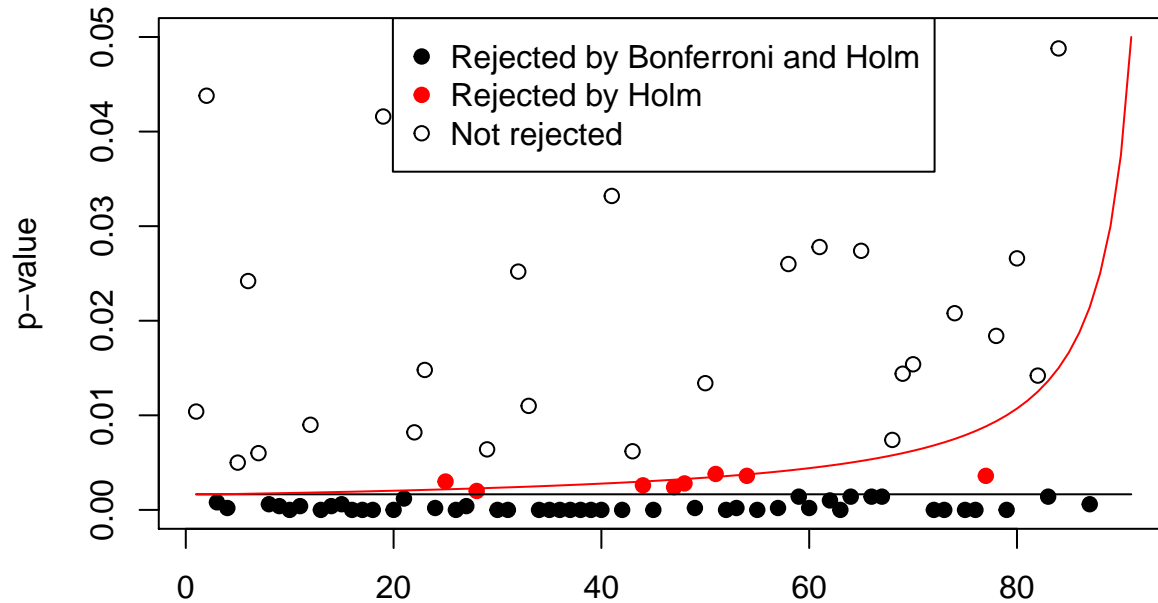
```

plot(pval, col=cols, pch=pchs, ylim=c(0, alpha),
     main="kFWER control using Bonferroni and Holm procedures",
     xlab="", ylab="p-value")
lines(attr(resBonf, 'tau'), cex=0.5, lty=1, col=1)
lines(attr(resHolm, 'tau'), cex=0.5, lty=1, col=2)

```

```
lgd <- c("Rejected by Bonferroni and Holm", "Rejected by Holm", "Not rejected")
legend("top", lgd, pch=c(19, 19, 1), col=c(1, 2, 1))
```

## kFWER control using Bonferroni and Holm procedures



- Proposer des améliorations de l'implémentation pour réduire le temps de calcul de la procédure.
- Comparer les temps de calculs et les résultats de la procédure originale et de celle implémentée en 5.

## Annexe

### Récupération 'manuelle' des scripts et des données

```
setwd("\\\\eden\\users\\pupitre\\Bureau\\kFWE")
load("kFWE.RData")
dat <- as.matrix(read.csv("ztwo.csv", header=FALSE))[1, ]
dat0 <- read.csv("znulltwo.csv", header=FALSE)
```

## Références

## References

- [1] Joseph P Romano, Azeem M Shaikh, and Michael Wolf. Formalized data snooping based on generalized error rates. *Econometric Theory*, (24):404–447, 2008.
- [2] Joseph P Romano and Michael Wolf. Control of Generalized Error Rates in Multiple Testing. *Ann. Statist.*, 35(4):1378–1408, 2007.