```r
## Load libraries
library(splines)
library(MASS)
library(xtable)

## Set the value of lambda
lambda <- 0.8

## Calculate maximum variance bound for different numbers of tests
nTests <- c(10, 100, 1000, 10000)
```

1 Probability of being a false positive as a linear function of time

```r
set.seed(1345)

##save the variance bound for each m
maxSm <- rep(NA, length(nTests))

for(m in 1:length(nTests))
  {
  ntest <- nTests[m]

  ## Set up the time vector and the probability of being null
  tme <- seq(-1,2,length=ntest)
  pi0 <- 1/4*tme+1/2

  ## Calculate a random variable indicating whether to draw
  ## the p-values from the null or alternative
  nullI <- rbinom(ntest,prob=pi0,size=1)> 0

  ## Sample the null P-values from U(0,1) and the alternatives
  ## from a beta distribution

  pValues <- rep(NA,ntest)
  pValues[nullI] <- runif(sum(nullI))
  pValues[!nullI] <- rbeta(sum(!nullI),1,50)

  ## Get the estimate

  y <- pValues > lambda
  glm1 <- glm(y ~ tme, x=TRUE)
```

```
  ##Get the variance bounds:
  zMat <- glm1$x
  S <- zMat%*%solve(t(zMat)%*%zMat)%*%t(zMat)
  maxSm[m] <-
    max(diag(S)/(4*(1-lambda)^2))
  }

xtable(matrix(maxSm, nrow=1), digits=3)

## % latex table generated in R 3.1.2 by xtable 1.7-4 package
## % Wed Dec 30 02:24:31 2015
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrrrr}
##   \hline
##  & 1 & 2 & 3 & 4 \\
##   \hline
## 1 & 2.159 & 0.246 & 0.025 & 0.002 \\
##    \hline
## \end{tabular}
## \end{table}
```

2 Probability of being a false positive as a smooth function of time

Linear term only:

```
set.seed(1345)

##save the variance bound for each m
maxSm <- rep(NA, length(nTests))

for(m in 1:length(nTests))
  {
  ntest <- nTests[m]

  ## Set up the time vector and the probability of being null
  tme <- seq(-1,2,length=ntest)
  pi0 <- pnorm(tme)

  ## Calculate a random variable indicating whether to draw
  ## the p-values from the null or alternative
  nullI <- rbinom(ntest,prob=pi0,size=1)> 0
```

```
  ## Sample the null P-values from U(0,1) and the alternatives
  ## from a beta distribution

  pValues <- rep(NA,ntest)
  pValues[nullI] <- runif(sum(nullI))
  pValues[!nullI] <- rbeta(sum(!nullI),1,50)

  ## Get the estimate

  y <- pValues > lambda
  glm1 <- glm(y ~ tme, x=TRUE)

  ##Get the variance bounds:
  zMat <- glm1$x
  S <- zMat%*%solve(t(zMat)%*%zMat)%*%t(zMat)
  maxSm[m] <-
    max(diag(S)/(4*(1-lambda)^2))
  }

xtable(matrix(maxSm, nrow=1), digits=3)

## % latex table generated in R 3.1.2 by xtable 1.7-4 package
## % Wed Dec 30 02:24:34 2015
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrrrr}
##   \hline
##  & 1 & 2 & 3 & 4 \\
##   \hline
## 1 & 2.159 & 0.246 & 0.025 & 0.002 \\
##    \hline
## \end{tabular}
## \end{table}
```

B-splines with 3 degrees of freedom:

```
set.seed(1345)

##save the variance bound for each m
maxSm <- rep(NA, length(nTests))

for(m in 1:length(nTests))
  {
  ntest <- nTests[m]
```

```r
  ## Set up the time vector and the probability of being null
  tme <- seq(-1,2,length=ntest)
  pi0 <- pnorm(tme)

  ## Calculate a random variable indicating whether to draw
  ## the p-values from the null or alternative
  nullI <- rbinom(ntest,prob=pi0,size=1)> 0

  ## Sample the null P-values from U(0,1) and the alternatives
  ## from a beta distribution

  pValues <- rep(NA,ntest)
  pValues[nullI] <- runif(sum(nullI))
  pValues[!nullI] <- rbeta(sum(!nullI),1,50)

  ## Get the estimate

  y <- pValues > lambda
  glm1 <- glm(y ~ ns(tme,df=3), x=TRUE)

  ##Get the variance bounds:
  zMat <- glm1$x
  S <- zMat%*%solve(t(zMat)%*%zMat)%*%t(zMat)
  maxSm[m] <-
    max(diag(S)/(4*(1-lambda)^2))
  }


xtable(matrix(maxSm, nrow=1), digits=3)

## % latex table generated in R 3.1.2 by xtable 1.7-4 package
## % Wed Dec 30 02:24:38 2015
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrrrr}
##   \hline
##  & 1 & 2 & 3 & 4 \\
##   \hline
## 1 & 4.697 & 0.726 & 0.076 & 0.008 \\
##    \hline
## \end{tabular}
## \end{table}
```

3 Probability of being a false positive as a sine + step function

3 degrees of freedom on the B-spline:

```
set.seed(1345)

##save the variance bound for each m
maxSm <- rep(NA, length(nTests))

for(m in 1:length(nTests))
  {
  ntest <- nTests[m]

  ## Set up the time vector and the probability of being null
  tme1 <- seq(-1*pi,2*pi,length=ntest)
  tme2 <- rep(1:0, each=ntest/2)
  pi0 <- 1/4*sin(tme1) + tme2/4 + 1/2

  ## Calculate a random variable indicating whether to draw
  ## the p-values from the null or alternative
  nullI <- rbinom(ntest,prob=pi0,size=1)> 0

  ## Sample the null P-values from U(0,1) and the alternatives
  ## from a beta distribution

  pValues <- rep(NA,ntest)
  pValues[nullI] <- runif(sum(nullI))
  pValues[!nullI] <- rbeta(sum(!nullI),1,50)

  ## Get the estimate

  y <- pValues > lambda
  glm1 <- glm(y ~ ns(tme1,df=3) + tme2, x=TRUE)

  ##Get the variance bounds:
  zMat <- glm1$x
  S <- zMat%*%ginv(t(zMat)%*%zMat)%*%t(zMat)
  maxSm[m] <-
    max(diag(S)/(4*(1-lambda)^2))
  }

xtable(matrix(maxSm, nrow=1), digits=3)

## % latex table generated in R 3.1.2 by xtable 1.7-4 package
## % Wed Dec 30 02:24:43 2015
```

```
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrrrr}
##   \hline
##  & 1 & 2 & 3 & 4 \\
##   \hline
## 1 & 4.798 & 0.755 & 0.079 & 0.008 \\
##    \hline
## \end{tabular}
## \end{table}
```

20 degrees of freedom on the B-spline:

```
set.seed(1345)

##save the variance bound for each m
maxSm <- rep(NA, length(nTests))

for(m in 1:length(nTests))
  {
  ntest <- nTests[m]

  ## Set up the time vector and the probability of being null
  tme1 <- seq(-1*pi,2*pi,length=ntest)
  tme2 <- rep(1:0, each=ntest/2)
  pi0 <- 1/4*sin(tme1) + tme2/4 + 1/2

  ## Calculate a random variable indicating whether to draw
  ## the p-values from the null or alternative
  nullI <- rbinom(ntest,prob=pi0,size=1)> 0

  ## Sample the null P-values from U(0,1) and the alternatives
  ## from a beta distribution

  pValues <- rep(NA,ntest)
  pValues[nullI] <- runif(sum(nullI))
  pValues[!nullI] <- rbeta(sum(!nullI),1,50)

  ## Get the estimate

  y <- pValues > lambda
  glm1 <- glm(y ~ ns(tme1,df=20) + tme2, x=TRUE)

  ##Get the variance bounds:
  zMat <- glm1$x
```

```
  S <- zMat%*%ginv(t(zMat)%*%zMat)%*%t(zMat)
  maxSm[m] <-
    max(diag(S)/(4*(1-lambda)^2))
  }

xtable(matrix(maxSm, nrow=1), digits=3)

## % latex table generated in R 3.1.2 by xtable 1.7-4 package
## % Wed Dec 30 02:24:51 2015
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrrrr}
##   \hline
##  & 1 & 2 & 3 & 4 \\
##   \hline
## 1 & 6.250 & 3.540 & 0.491 & 0.051 \\
##    \hline
## \end{tabular}
## \end{table}
```

Session info:

```
devtools::session_info()

## Session info -------------------------------------------------

##  setting  value
##  version  R version 3.1.2 (2014-10-31)
##  system   x86_64, mingw32
##  ui       RTerm
##  language (EN)
##  collate  English_United States.1252
##  tz       America/New_York
##  date     2015-12-30

## Packages -----------------------------------------------------

##  package  * version date       source
##  devtools   1.9.1   2015-09-11 CRAN (R 3.1.3)
##  digest     0.6.8   2014-12-31 CRAN (R 3.1.2)
##  evaluate   0.5.5   2014-04-29 CRAN (R 3.1.1)
##  formatR    1.0     2014-08-25 CRAN (R 3.1.2)
##  highr      0.4     2014-10-23 CRAN (R 3.1.2)
##  knitr    * 1.9     2015-01-20 CRAN (R 3.1.2)
##  MASS     * 7.3-35  2014-09-30 CRAN (R 3.1.2)
##  memoise    0.2.1   2014-04-22 CRAN (R 3.1.3)
##  stringr    0.6.2   2012-12-06 CRAN (R 3.1.1)
##  xtable   * 1.7-4   2014-09-12 CRAN (R 3.1.2)
```