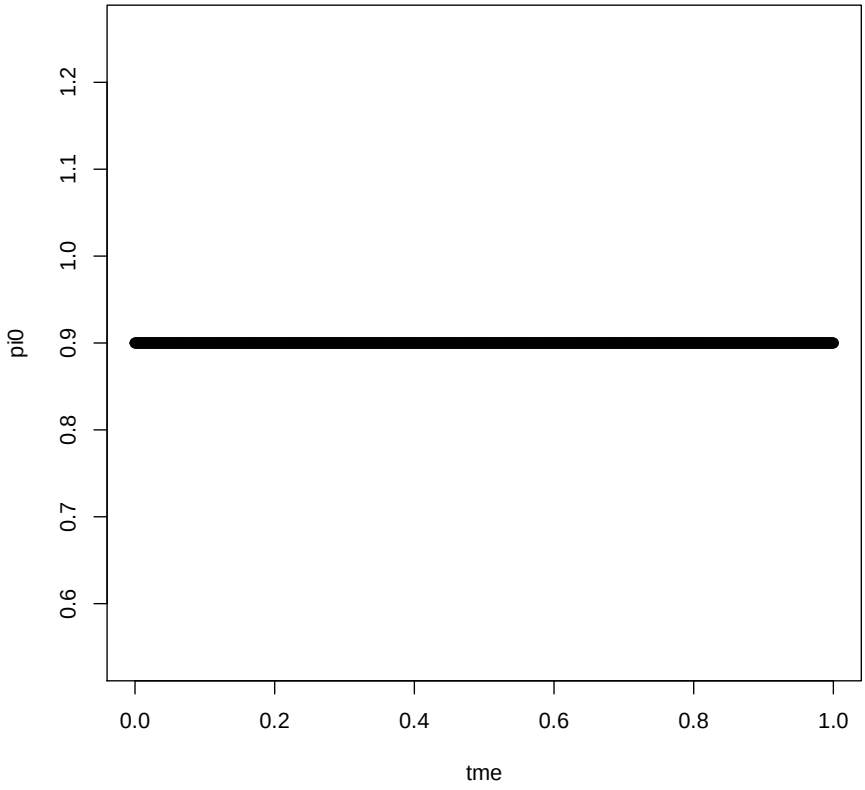
```
## Load libraries
library(splines)
library(MASS)

library(mnormt) ## for multivariate normal and t distributions
library(Matrix) ##for the bdiag function to create block-diagonal matrices

library(doParallel) ##to make cluster (on Windows)

## Loading required package:  foreach
## Loading required package:  iterators
## Loading required package:  parallel

library(foreach) ##to use foreach function that does the parallel processing
library(doRNG) ##for reproducible seeds when doing parallel processing

## Loading required package:  rngtools
## Loading required package:  pkgmaker
## Loading required package:  registry
##
## Attaching package:  'pkgmaker'
## The following object is masked from 'package:base':
##
##    isNamespaceLoaded

##Source functions
source("../functions.R")

## Define the number of tests
ntest <- 1000

## Set number of simulations
nSims <- 200
```

Do the simulations for a variety of alternative distributions:

```
altsGrid <- as.matrix(expand.grid(dist=c("z","t"),nrBlocks=c(10,20),corr=c(0.2,0.5,0.9)))
alts <- apply(altsGrid, 1, function(x){paste("alt",x[1],"large",x[2],x[3],sep="_")})
alts

##  [1] "alt_z_large_10_0.2" "alt_t_large_10_0.2"
##  [3] "alt_z_large_20_0.2" "alt_t_large_20_0.2"
##  [5] "alt_z_large_10_0.5" "alt_t_large_10_0.5"
##  [7] "alt_z_large_20_0.5" "alt_t_large_20_0.5"
##  [9] "alt_z_large_10_0.9" "alt_t_large_10_0.9"
## [11] "alt_z_large_20_0.9" "alt_t_large_20_0.9"
```

1 Probability of being a false positive is flat

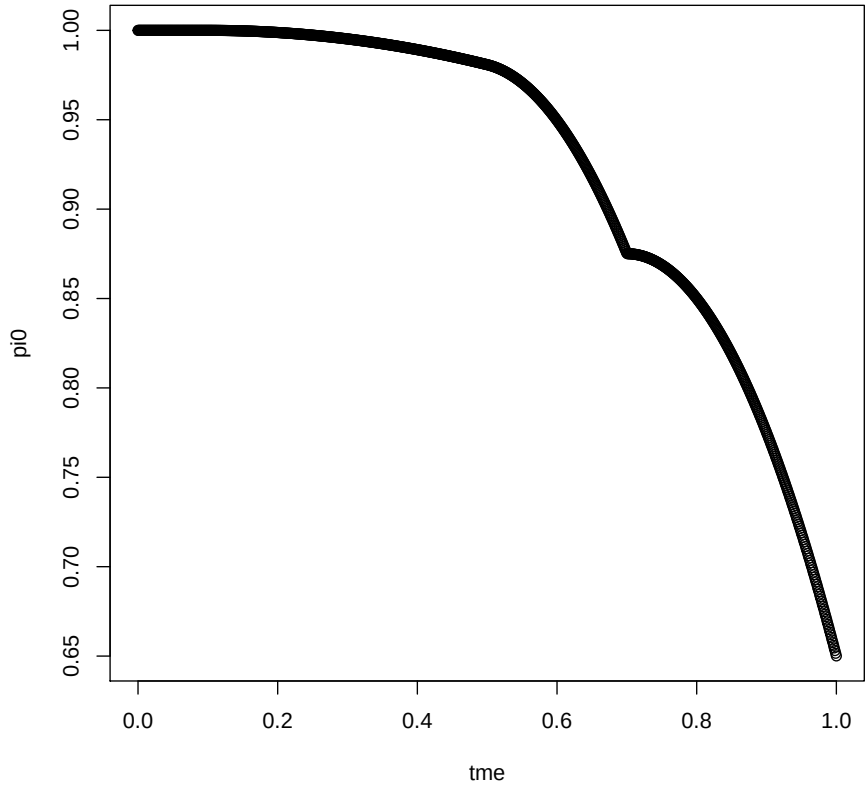
```
## Set up the time vector and the probability of being null
tme <- seq(0,1, length=ntest)
pi0 <- rep(0.9, length=ntest)

plot(pi0 ~ tme)

for(alt in alts)
{
  pValuesSims <- run_sims_alt_corr(alt, nSims, pi0)

  zValuesSims <- pValuesSims[,(2*ntest+1):(3*ntest)]
  nullHypSims <- pValuesSims[,(ntest+1):(2*ntest)]
  pValuesSims <- pValuesSims[,1:ntest]

  ##save results
  save(file=paste(alt, "simResults_1.RData",sep="/"),
       list=c("pi0", "tme", "nullHypSims","pValuesSims","zValuesSims"))
}
```



2 Probability of being a false positive is smooth in one variable

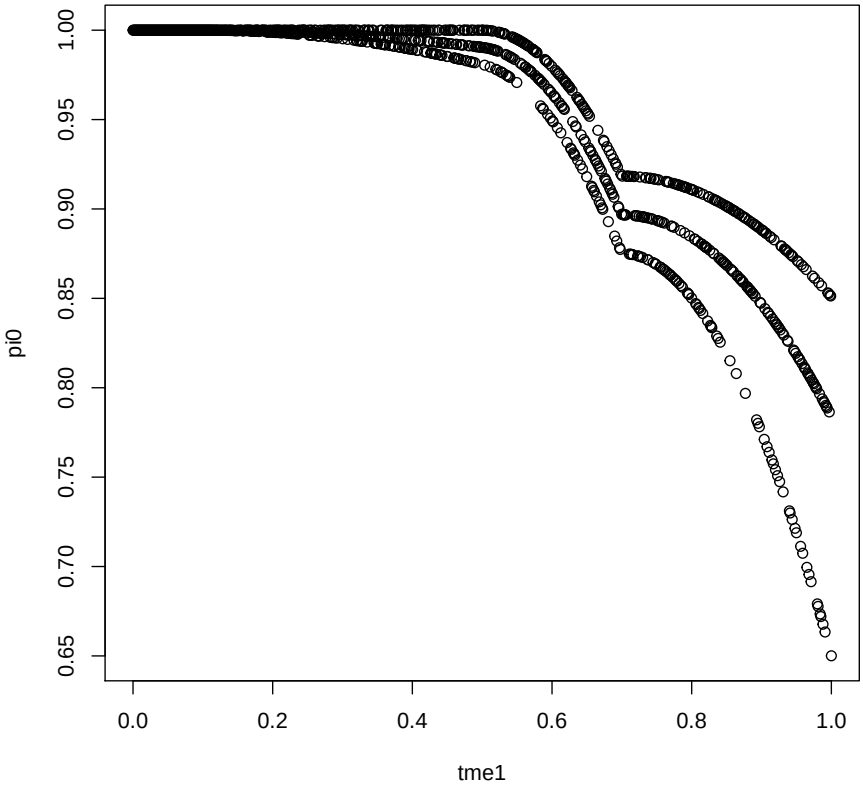
```
## Set up the time vector and the probability of being null
tme <- seq(0,1, length=ntest)
pi0 <- fSingle(tme)

plot(pi0 ~ tme)

for(alt in alts)
{
  pValuesSims <- run_sims_alt_corr(alt, nSims, pi0)

  dim(pValuesSims)
```

```
  zValuesSims <- pValuesSims[,(2*ntest+1):(3*ntest)]
  nullHypSims <- pValuesSims[,(ntest+1):(2*ntest)]
  pValuesSims <- pValuesSims[,1:ntest]

  ##save results
  save(file=paste(alt, "simResults_2.RData",sep="/"),
       list=c("pi0", "tme", "nullHypSims","pValuesSims","zValuesSims"))
}
```



3 Probability of being a false positive is smooth in one variable within levels of second variable

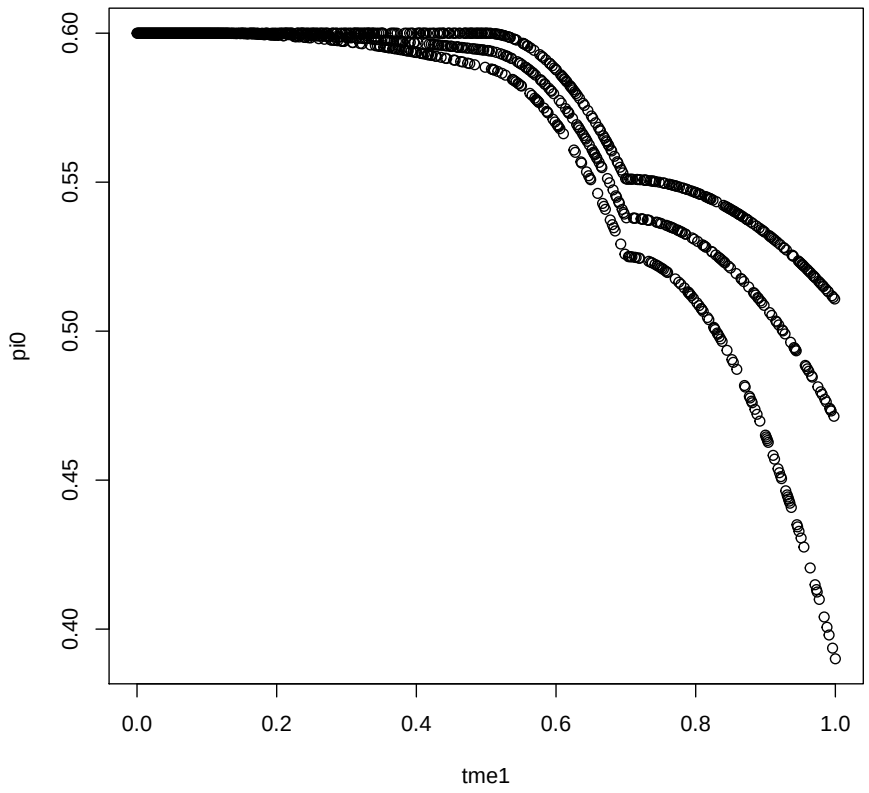
```r
## Set up the time vector and the probability of being null
tme1 <- seq(0,1,length=ntest)
tme2cont <- runif(ntest,0,0.5)
set.seed(309441)
tme2 <- rep(NA, ntest)
tme2[tme2cont < 0.127] <- 1
tme2[tme2cont >= 0.127] <- 2
tme2[tme2cont >= 0.302] <- 3
pi0 <- f(tme1, tme2)
range(pi0)

## [1] 0.6500664 1.0000000

plot(pi0 ~ tme1)

for(alt in alts)
{
  pValuesSims <- run_sims_alt_corr(alt, nSims, pi0)

  dim(pValuesSims)

  zValuesSims <- pValuesSims[,(2*ntest+1):(3*ntest)]
  nullHypSims <- pValuesSims[,(ntest+1):(2*ntest)]
  pValuesSims <- pValuesSims[,1:ntest]

  ##save results
  save(file=paste(alt, "simResults_3.RData",sep="/"),
       list=c("pi0", "tme1", "tme2", "nullHypSims","pValuesSims","zValuesSims"))
}
```



4 Probability of being a false positive is smooth in one variable within levels of second variable - lower priors

```
## Set up the time vector and the probability of being null
tme1 <- seq(0,1,length=ntest)
tme2cont <- runif(ntest,0,0.5)
set.seed(309441)
tme2 <- rep(NA, ntest)
tme2[tme2cont < 0.127] <- 1
tme2[tme2cont >= 0.127] <- 2
tme2[tme2cont >= 0.302] <- 3
pi0 <- 0.6*f(tme1, tme2)
```

```r
range(pi0)

## [1] 0.3900398 0.6000000

plot(pi0 ~ tme1)

for(alt in alts)
{
  pValuesSims <- run_sims_alt_corr(alt, nSims, pi0)

  dim(pValuesSims)

  zValuesSims <- pValuesSims[,(2*ntest+1):(3*ntest)]
  nullHypSims <- pValuesSims[,(ntest+1):(2*ntest)]
  pValuesSims <- pValuesSims[,1:ntest]

  ##save results
  save(file=paste(alt, "simResults_4.RData",sep="/"),
       list=c("pi0", "tme1", "tme2", "nullHypSims","pValuesSims","zValuesSims"))
}
```



Session info:

```
devtools::session_info()

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

##  setting  value
##  version  R version 3.4.0 (2017-04-21)
##  system   x86_64, mingw32
##  ui       RTerm
##  language (EN)
##  collate  English_United States.1252
##  tz       America/New_York
##  date     2017-06-01

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

##  package    * version date       source
```

```
##  codetools    0.2-15  2016-10-05 CRAN (R 3.4.0)
##  devtools     1.12.0  2016-12-05 CRAN (R 3.4.0)
##  digest       0.6.12  2017-01-27 CRAN (R 3.4.0)
##  doParallel * 1.0.10  2015-10-14 CRAN (R 3.4.0)
##  doRNG      * 1.6.6   2017-04-10 CRAN (R 3.4.0)
##  evaluate     0.10    2016-10-11 CRAN (R 3.4.0)
##  foreach    * 1.4.3   2015-10-13 CRAN (R 3.4.0)
##  highr        0.6     2016-05-09 CRAN (R 3.4.0)
##  iterators  * 1.0.8   2015-10-13 CRAN (R 3.4.0)
##  knitr      * 1.15.1  2016-11-22 CRAN (R 3.4.0)
##  lattice      0.20-35 2017-03-25 CRAN (R 3.4.0)
##  magrittr     1.5     2014-11-22 CRAN (R 3.4.0)
##  MASS       * 7.3-47  2017-02-26 CRAN (R 3.4.0)
##  Matrix     * 1.2-9   2017-03-14 CRAN (R 3.4.0)
##  memoise      1.1.0   2017-04-21 CRAN (R 3.4.0)
##  mnormt     * 1.5-5   2016-10-15 CRAN (R 3.4.0)
##  pkgmaker   * 0.22    2014-05-14 CRAN (R 3.4.0)
##  registry   * 0.3     2015-07-08 CRAN (R 3.4.0)
##  rngtools   * 1.2.4   2014-03-06 CRAN (R 3.4.0)
##  stringi      1.1.5   2017-04-07 CRAN (R 3.4.0)
##  stringr      1.2.0   2017-02-18 CRAN (R 3.4.0)
##  withr        1.0.2   2016-06-20 CRAN (R 3.4.0)
##  xtable       1.8-2   2016-02-05 CRAN (R 3.4.0)
```