R Notebook for L0TF

2023-04-10

1 Load Simulation Functions

```
library(genlasso)
library("AMIAS")
```

2 Toy Example for L0TF paper

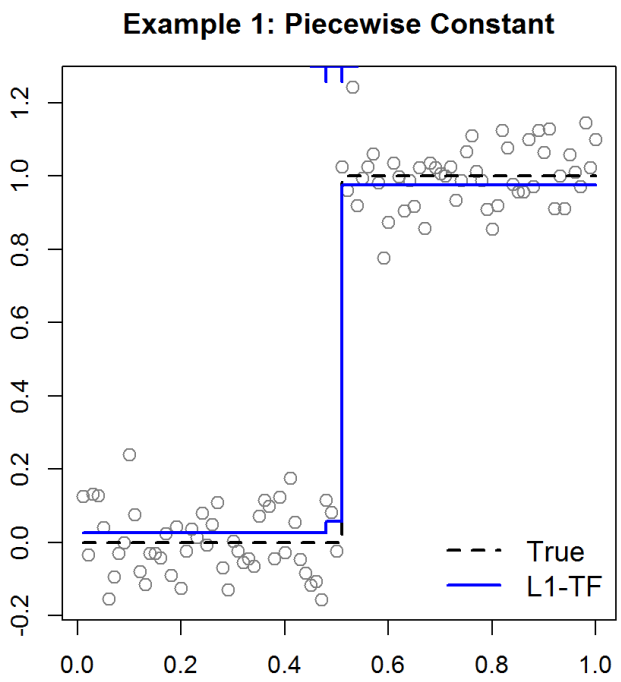
```
# ------------------------------------
# Toy Example: Piecewise Constant/Linear Simulation
# ------------------------------------
ToyEx <- function(n, sigma=0.1, q=0, seed=NA) {
  tau = 0.5
  if (!is.na(seed)) set.seed(seed)
  x = seq(1/n, 1,length.out = n)
  if (q==0){
    y0 = 0*x; y0[x>tau] = 1
  }
  if (q==1) {
    y0 = 2*(tau-x); y0[x>tau] = 2*(x[x>tau]-tau)
  }
  y = y0 + sigma*rnorm(n)
  return(list(y = y, x = x, y0 = y0, tau = tau))
}

# ------------------------------------
# Simu Example: Piecewise Constant/Linear Simulation
#     Equal-spaced knots or Random Knots
# ------------------------------------
SimuEx <- function(n, sigma=0.1, q=0, nknot=4, seed=NA, RandKnot=FALSE, AdaKnot=FALSE) {
  if (!is.na(seed)) set.seed(seed)
  x = seq(1/n, 1,length.out = n)
  A=round(seq(0, n, length.out=nknot+2))[seq(2,nknot+1)]
  if(RandKnot) A = sort(sample(seq(6, n-5, 5), nknot))
  if(AdaKnot) A = round(seq(1, sqrt(n), length.out=nknot+2)^2)[seq(2,nknot+1)]
  tau = x[A]
  tau1 = c(0, tau, 1)
  if (q==0){
    aa = 1-seq(1,nknot+1)%%2
    y0 = 0*x
    for (j in 1:(nknot+1)) y0[x>tau1[j] & x<=tau1[j+1]] = aa[j]
  }
  if (q==1) {
    aa = 2*(-1)^seq(1,nknot+1)
    phi = rep(1, n)
    for (j in 1:(nknot+1)) phi = cbind(phi,pmin(pmax(x-tau1[j], 0), tau1[j+1]-tau1[j]))
    y0 = phi%*%c(0.5+1/(nknot+1),aa)
  }
  y = y0 + sigma*rnorm(n)
  return(list(y = y, x = x, y0 = y0, tau = tau, SetA = A))
}
```

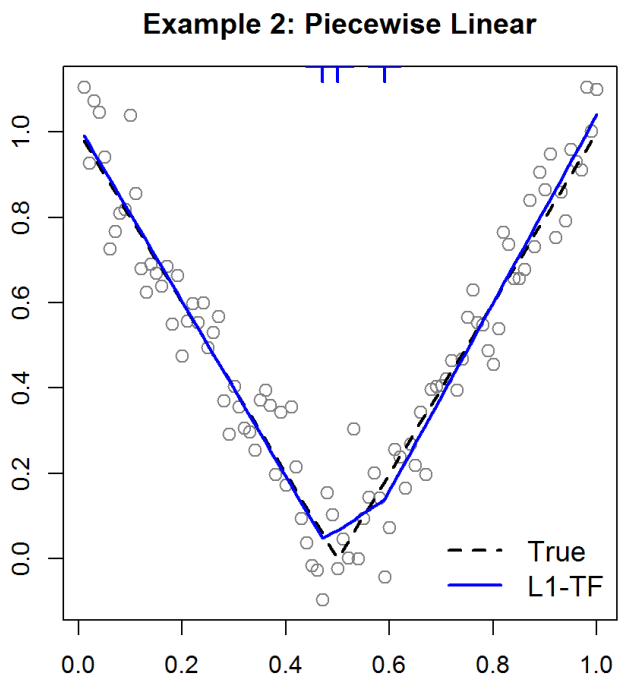
2.1 Figure 2: Toy examples of ℓ_1-TF results for piecewise constant

and piecewise linear cases

```
# ------------------------------------
# Wrapped-up L1TF
# ------------------------------------
L1TF <- function(data, n, q) {
  resL1  <- trendfilter(pos = data$x, y = data$y, ord=q)
  sighat <- median(abs(diff(data$y, diff = q+1)))/(qnorm(3/4)*sqrt(choose(2*(q+1), q+1)))
  bicL1 <- apply(resL1$beta, 2, function(beta) sum((data$y-beta)^2)/sighat^2) + 2*log(n)*resL1$df
  #plot(bicL1)
  betaL1 <- resL1$beta[,which.min(bicL1)]
  knotL1 = data$x[which(abs(diff(betaL1, diff=q+1))>1e-5)+1]
  return(list(beta=betaL1, knot=knotL1))
}



RunPlot = function(data, resL1, q){
  ltype = ifelse(q==0, "s", "l")
  plot(data$x, data$y, type='p', col='grey50', xlab="", ylab="", cex=1.2)
  lines(data$x, data$y0, col=1, lty=2, lwd=2, type=ltype)
  lines(data$x, resL1$beta, col=4, lwd=2, type=ltype)
  points(resL1$knot, rep(par("usr")[4], length(resL1$knot)), pch=3, cex=2, col=4, lwd=2)
  legend("bottomright", c("True", "L1-TF"),
         lty=c(2,1), col=c(1,4), lwd=2,
         cex=1.2, bty="n")
}
```

```
## Run L1TF simulation with seed=1 for q=0 and q=1
# png("../L0TF_IJOC_Revision/l0tf0513/ToyExample_submit.png", pointsize = 8, width=850, height=400, res = 120)
par(mfrow=c(1,2),mar=c(3,3,3,3))
n=100; q=0; sigma=0.1; seed=0;
data = ToyEx(n=n, q=q, sigma=sigma, seed=seed)
resL1 = L1TF(data, n, q)
RunPlot(data, resL1, q)
title(main="Example 1: Piecewise Constant")
n=100; q=1; sigma=0.1; seed=0;
data = ToyEx(n=n, q=q, sigma=sigma, seed=seed)
resL1 = L1TF(data, n, q)
RunPlot(data, resL1, q)
title(main="Example 2: Piecewise Linear")
```

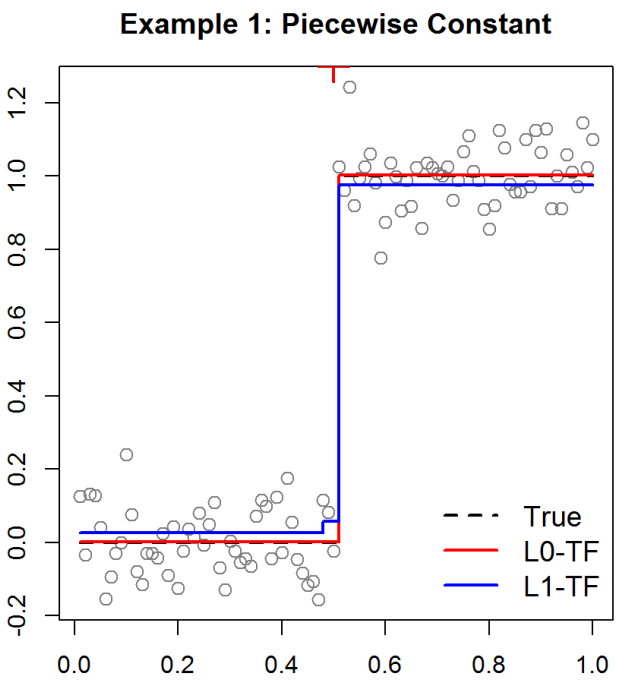


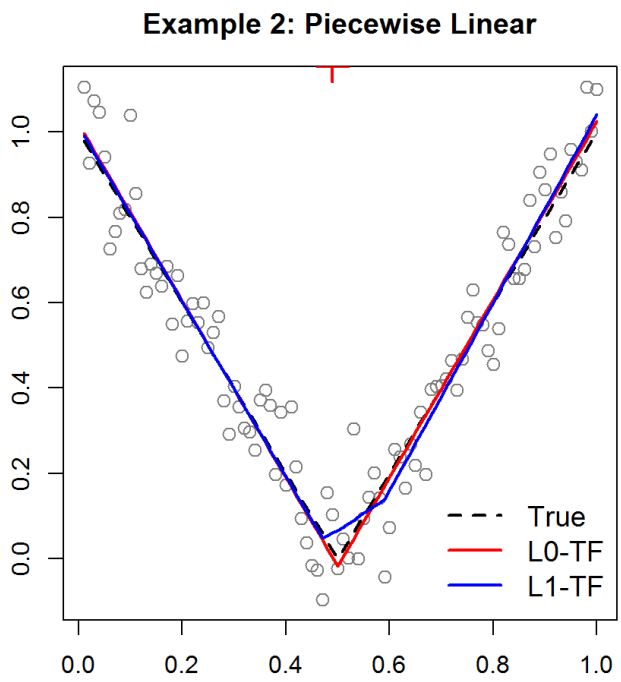



```
#  dev.off()
```

2.2 Figure 4: Toy examples from Figure 2, refitted by ℓ_0-TF with AMIAS algorithm

```
RunPlot = function(data, resL0, resL1, q){
  ltype = ifelse(q==0, "s", "l")
  plot(data$x, data$y, type='p', col='grey50', xlab="", ylab="", cex=1.2)
  lines(data$x, data$y0, col=1, lty=2, lwd=2, type=ltype)
  lines(data$x, resL0$alpha, col=2, lwd=2, type=ltype)
  lines(data$x, resL1$beta, col=4, lwd=2, type=ltype)
  points(data$x[resL0$A], rep(par("usr")[4], length(resL0$A)), pch=3, cex=2, col=2, lwd=2)
  legend("bottomright", c("True", "L0-TF", "L1-TF"),
         lty=c(2,1,1), col=c(1,2,4), lwd=2,
         cex=1.2, bty="n")
}

# png("../L0TF_IJOC_Revision/l0tf0513/ToyExample2.png", pointsize = 8, width=850, height=400, res = 120)
par(mfrow=c(1,2),mar=c(3,3,3,3))

# Toy Piecewise constant/linear: one knot only
n=100; q=0; sigma=0.1; seed=0; nknot=1
data = ToyEx(n=n, q=q, sigma=sigma, seed=seed)
resL1 = L1TF(data, n, q)

resL0 = amias(data$y, D_type="tf0", k=nknot)
RunPlot(data, resL0, resL1, q)
title(main="Example 1: Piecewise Constant")

n=100; q=1; sigma=0.1; seed=0; nknot=1
data = ToyEx(n=n, q=q, sigma=sigma, seed=seed)
resL1 = L1TF(data, n, q)
resL0 = amias(data$y, D_type="tfq", q=q, k=nknot)
RunPlot(data, resL0, resL1, q)
title(main="Example 2: Piecewise Linear")
```



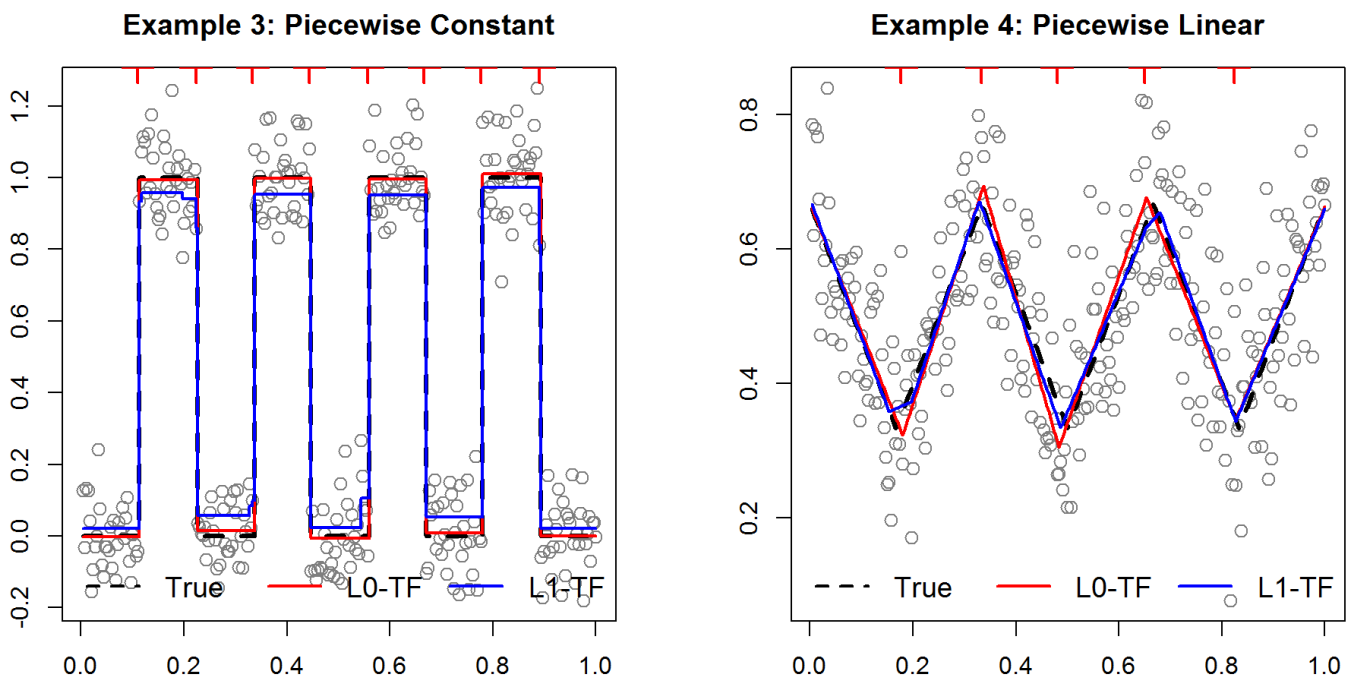



```
# dev.off()
```

2.3 Figure 5: Simulation examples of piecewise constant and

piecewise linear trends.

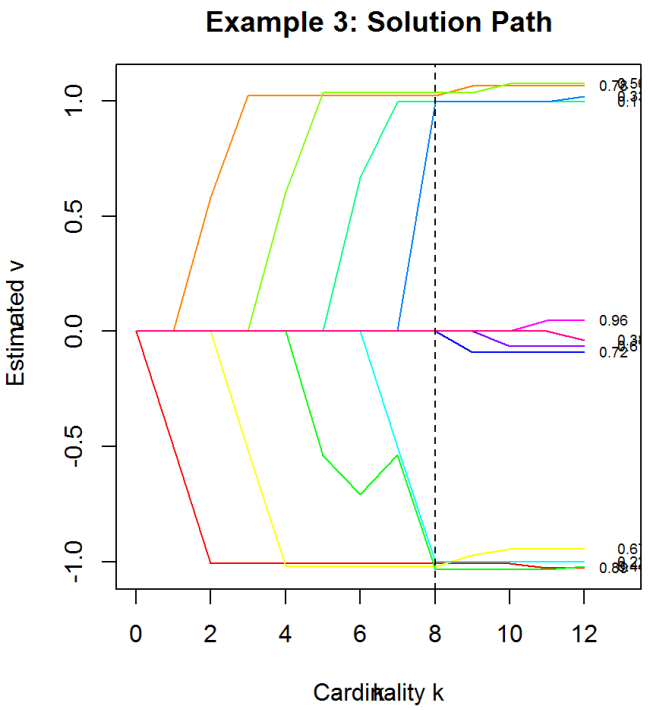
```
# ------------------------------------
# Run Piecewise-Constant/Linear: L1TF vs. L0TF
# Seed = 1
# Equal-spaced knots: 8 (q=0) and 5 (q=1)
# ------------------------------------
# png("SimuEx1.png", pointsize = 8, width=850, height=400, res = 120)
par(mfrow=c(1,2),mar=c(3,3,3,3))
# Piecewise constant case
n=300; sigma=0.1; q=0; nknot=8;
data = SimuEx(n=n, sigma=sigma, q=q, nknot=nknot, seed=0)
resL0 = samias(as.numeric(data$y), D_type="tf0", kmax=nknot+4)
resL1 = L1TF(data, n, q)

ltype = ifelse(q==0, "s", "l")
plot(data$x, data$y, type='p', col='grey50', xlab="", ylab="", cex=1.2)
lines(data$x, data$y0, col=1, lty=2, lwd=3, type=ltype)
lines(data$x, resL0$alpha, col=2, lwd=2, type=ltype)
lines(data$x, resL1$beta, col=4, lwd=2, type=ltype)
points(data$x[resL0$A], rep(par("usr")[4], length(resL0$A)), pch=3, cex=2, col=2, lwd=2)
legend("bottom", c("True", "L0-TF", "L1-TF"), lty=c(2,1,1), col=c(1,2,4), lwd=c(2,2,2),
       horiz=T,  bty="n",
       cex=1.2)
title(main="Example 3: Piecewise Constant")

n=300; sigma=0.1; q=1; nknot=5;
data = SimuEx(n=n, sigma=sigma, q=q, nknot=nknot, seed=0)
resL0 = samias(as.numeric(data$y), D_type="tfq", q=q, kmax=nknot+4, adjust = TRUE )
resL1 = L1TF(data, n, q)
ltype = ifelse(q==0, "s", "l")
plot(data$x, data$y, type='p', col='grey50', xlab="", ylab="", cex=1.2)
lines(data$x, data$y0, col=1, lty=2, lwd=3, type=ltype)
lines(data$x, resL0$alpha, col=2, lwd=2, type=ltype)
lines(data$x, resL1$beta, col=4, lwd=2, type=ltype)
points(data$x[resL0$A], rep(par("usr")[4], length(resL0$A)), pch=3, cex=2, col=2, lwd=2)
legend("bottom", c("True", "L0-TF", "L1-TF"), lty=c(2,1,1), col=c(1,2,4), lwd=c(2,2,2),
       horiz=T,  bty="n", cex=1.2)
title(main="Example 4: Piecewise Linear")
```

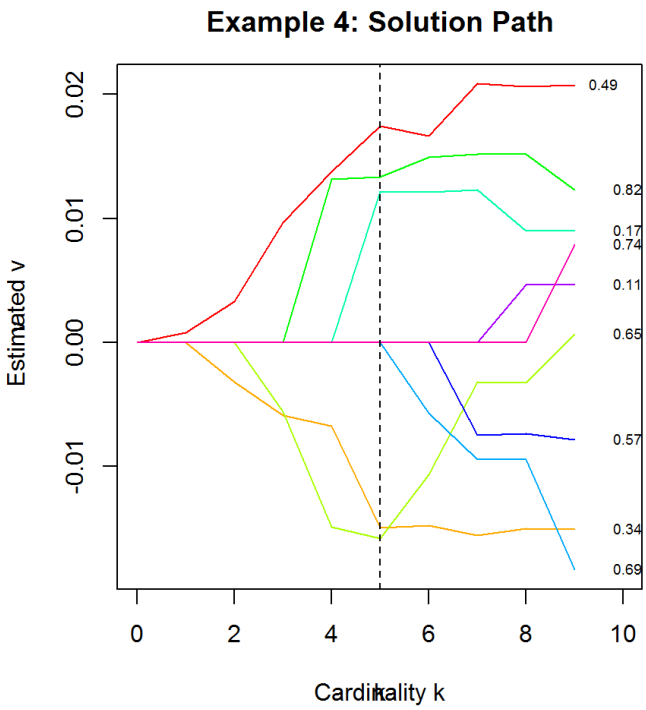


```
# dev.off()
```

2.4 Figure 6: Solution paths for the estimated primal variables subject to an ℓ_0 penalty

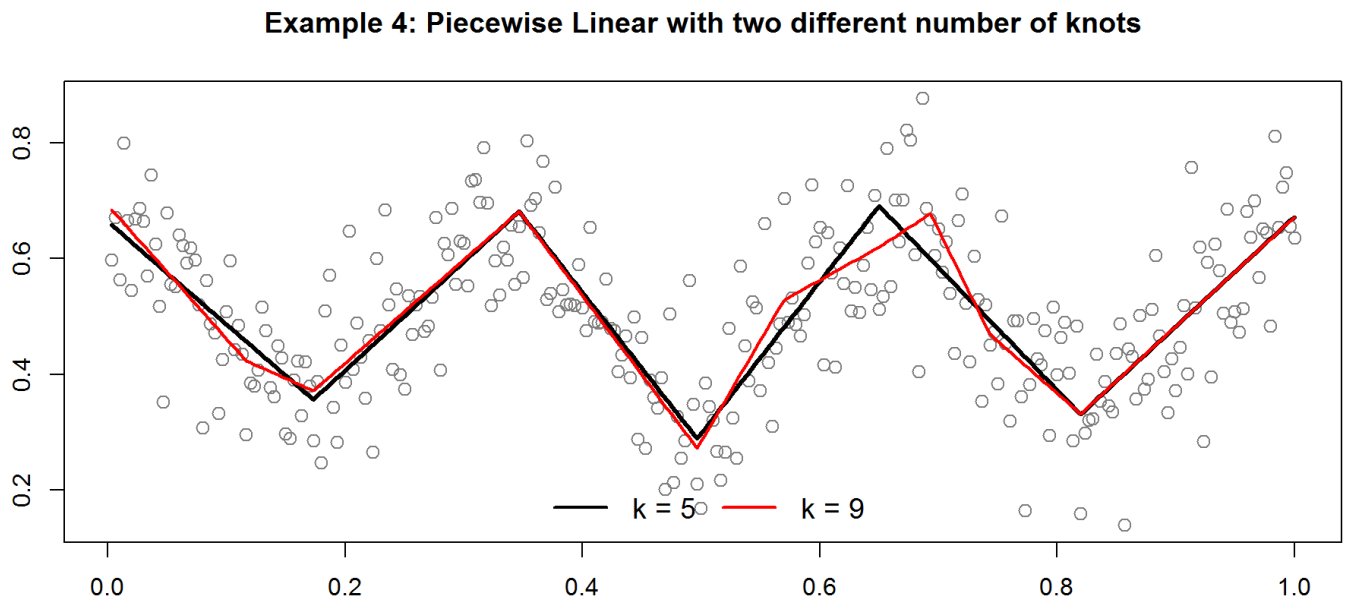
```
par(mfrow=c(1,2),mar=c(4,4,3,3))
n=300; sigma=0.1; q=0; nknot=8; seed=1
data = SimuEx(n=n, sigma=sigma, q=q, nknot=nknot, seed=seed)
sighat = median(abs(diff(data$y, diff = q+1)))/(qnorm(3/4)*sqrt(choose(2*(q+1), q+1)))

resL0 = samias(as.numeric(data$y), D_type= "tf0", kmax=nknot+4, eps=(0.92*sighat)^2, adjust = TRUE)
plot(resL0, type="vpath")
title(main="Example 3: Solution Path", xlab="k", ylab = "v")

n=300; sigma=0.1; q=1; nknot=5; seed=1
data = SimuEx(n=n, sigma=sigma, q=q, nknot=nknot, seed=seed)
sighat = median(abs(diff(data$y, diff = q+1)))/(qnorm(3/4)*sqrt(choose(2*(q+1), q+1)))

resL0 = samias(as.numeric(data$y), D_type= "tfq", q=q, kmax=nknot+4, eps=(0.96*sighat)^2, adjust = TRUE)
plot(resL0, type="vpath")
title(main="Example 4: Solution Path", xlab="k", ylab = "v")
```





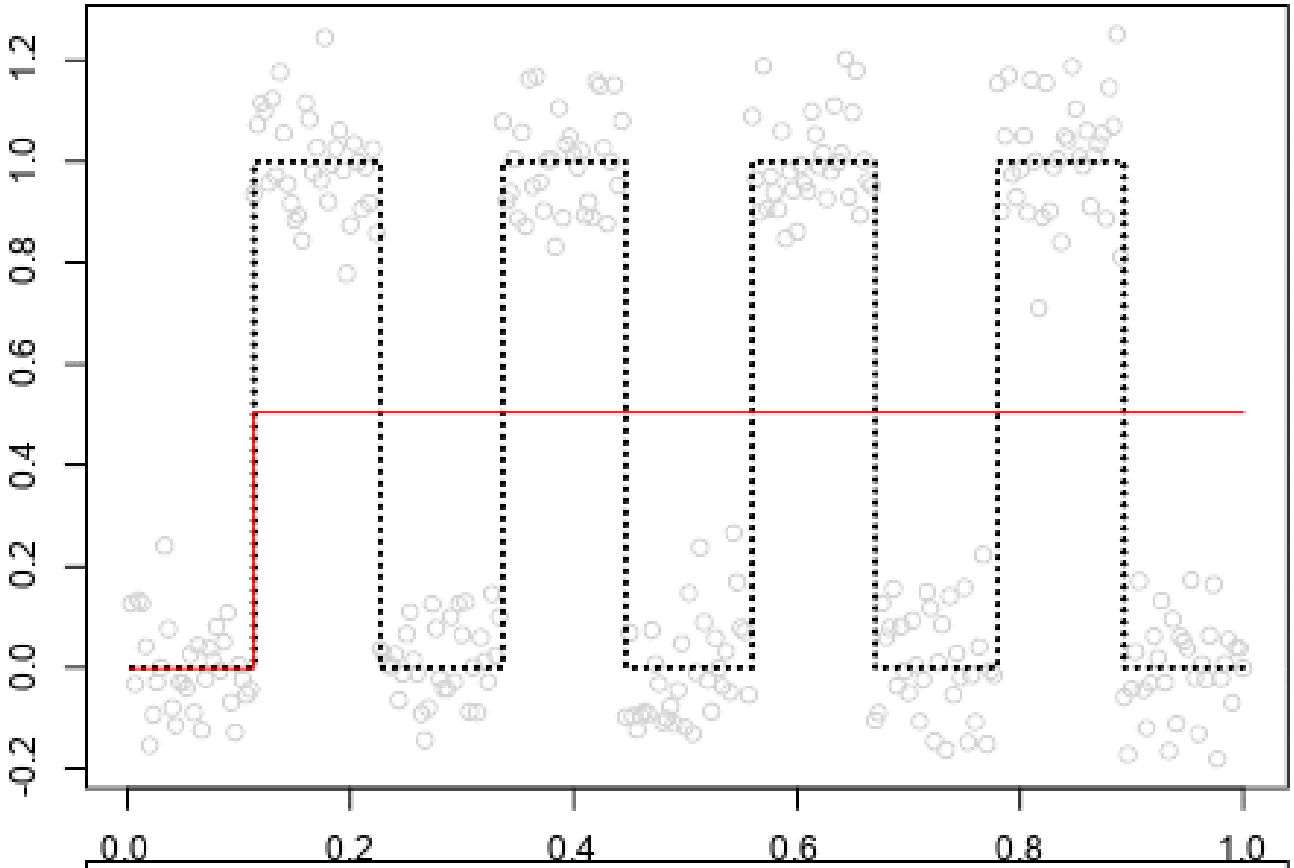
2.5 Figure 6: L0-TF estimates for nknot=5 and nknot=9

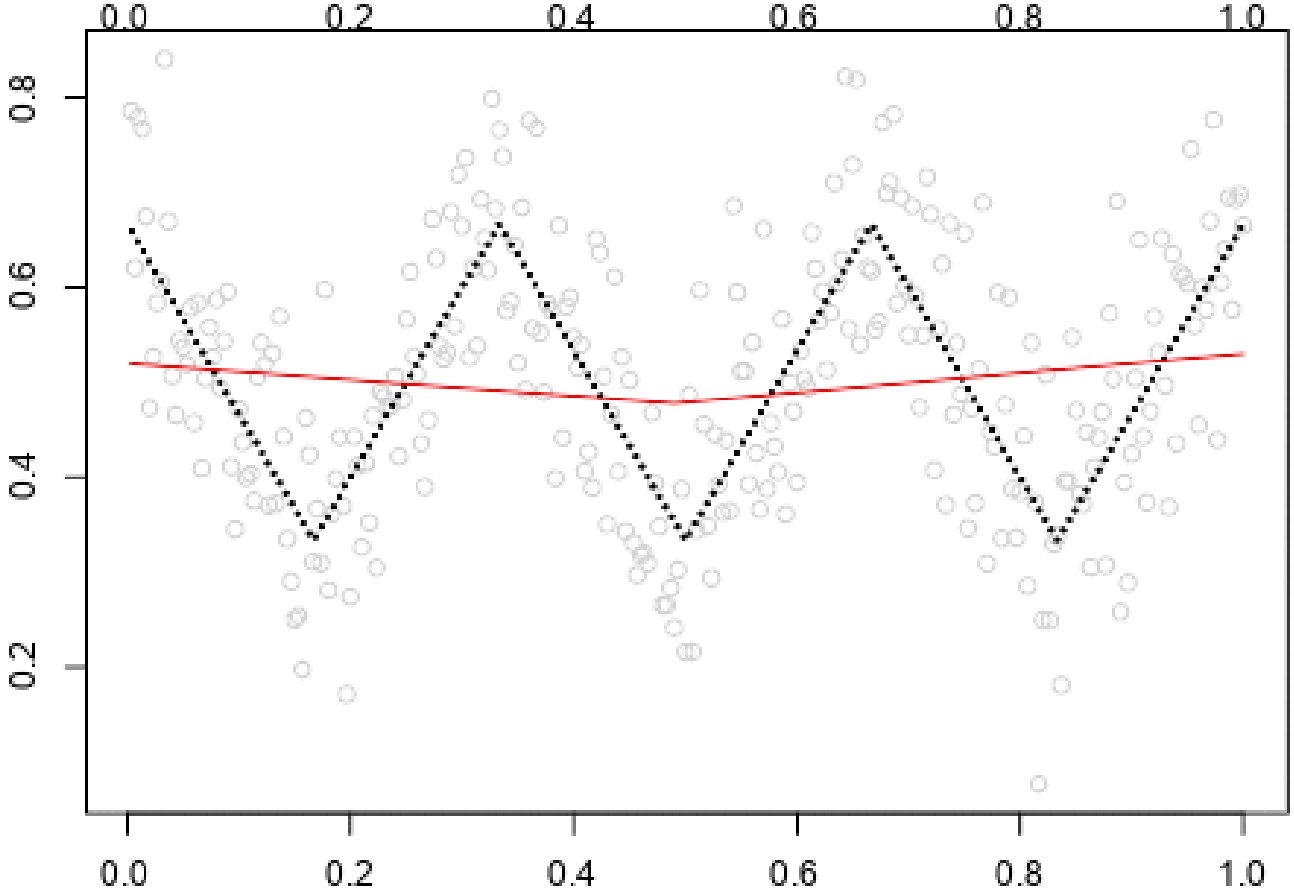
```
# png("../L0TF_IJOC_Revision/l0tf0513/SimuEx1_2.png", pointsize = 8, width=850, height=400, res = 120)
n=300; sigma=0.1; q=1; nknot=5; seed=1
data = SimuEx(n=n, sigma=sigma, q=q, nknot=nknot, seed=seed)
sighat = median(abs(diff(data$y, diff = q+1)))/(qnorm(3/4)*sqrt(choose(2*(q+1), q+1)))

resL0 = samias(as.numeric(data$y), D_type= "tfq", q=q, kmax=nknot+4, eps=(0.96*sighat)^2, adjust = TRUE)
ltype = ifelse(q==0, "s", "l")
plot(data$x, data$y, type='p', col='grey50', xlab="", ylab="", cex=1.2)
lines(data$x, resL0$alpha.all[,5], col=1, lty=1, lwd=3, type=ltype)
lines(data$x, resL0$alpha.all[,9], col=2, lwd=2, type=ltype)

legend("bottom", c("k = 5", "k = 9"), lty=c(1,1), col=c(1,2), lwd=c(2,2),
       horiz=T,  bty="n", cex=1.2)
title(main="Example 4: Piecewise Linear with two different number of knots")
```

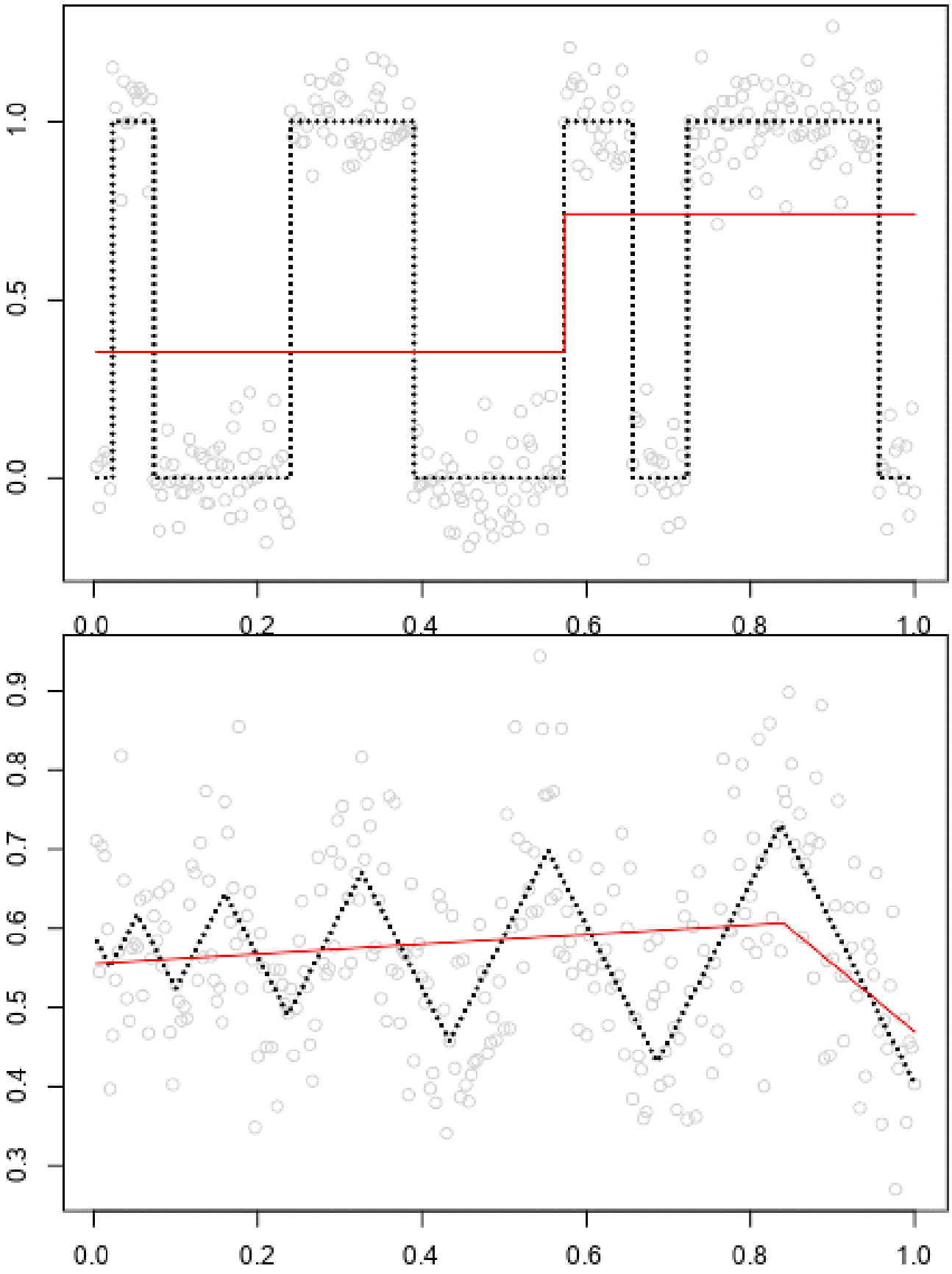



```
# dev.off()
```

2.6 Additional GIF figures showing the solution at each step





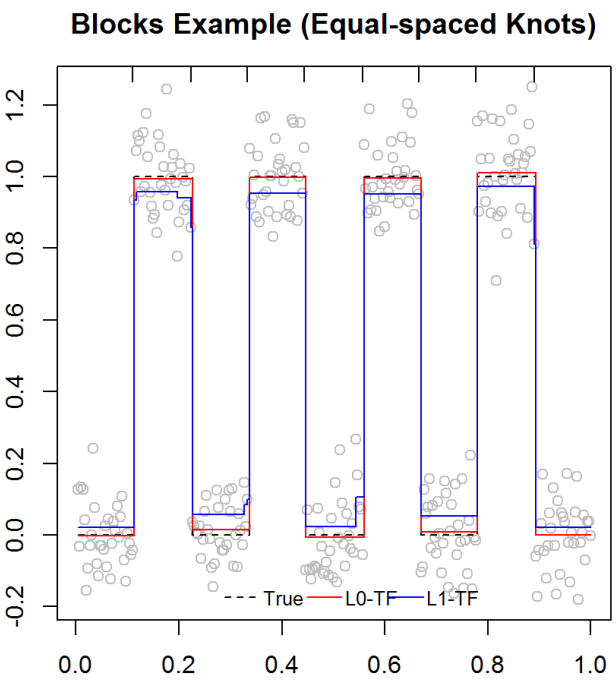


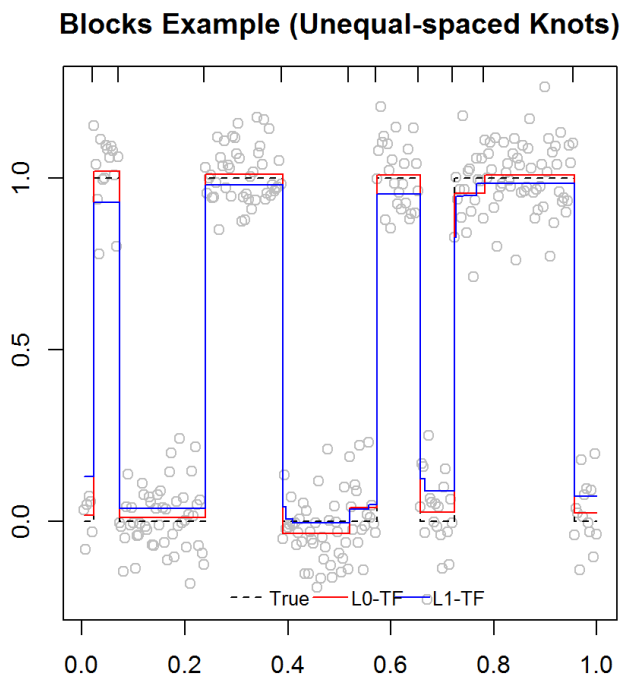
3 Comparing L0TF to L1TF via more simulated data

```
# ------------------------------------
# Wrap-up functions for running L1TF and L0TF
# ------------------------------------
RunL0L1 <- function(data, n, q, nknot=length(data$tau), TopMethod="pks"){
  if(q==0){
     resL0 = samias(as.numeric(data$y), D_type="tf0", kmax=nknot+4, tmax=10, adjust = TRUE)
    }else{
     resL0 = samias(as.numeric(data$y), D_type="tfq", q = q, kmax=nknot+4, tmax=10, adjust = TRUE)
    }


  resL1 = L1TF(data, n, q)

  return(list(L0fit = resL0$alpha, L1fit = resL1$beta, L0knot=resL0$A/n))
}
```

3.1 Piecewise Constant Case

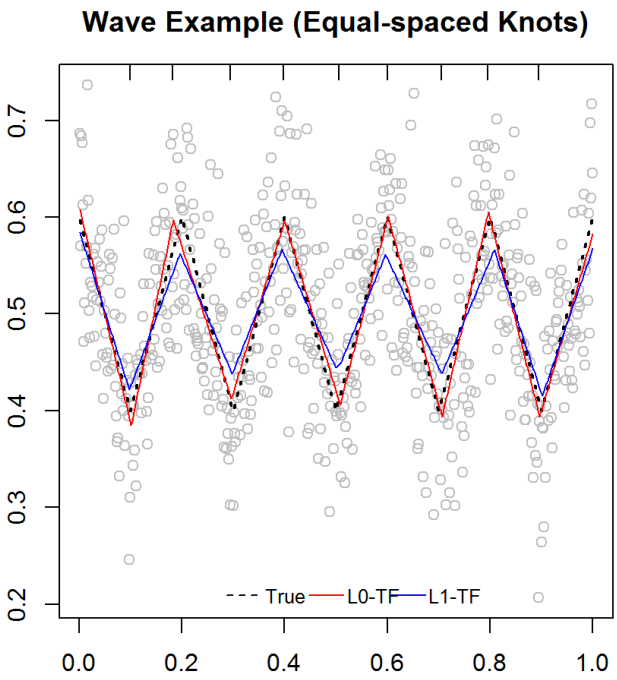
```
# ------------------------------------
# Run Piecewise-constant: L1TF vs. L0TF
# Seed = 1
# Equal-/Unequal-spaced knots
# Random Knots for Piecewise-constant, seed=1
# ------------------------------------
par(mfrow=c(1,2),mar=c(3,3,3,3))
# Piecewise constant case
n=300; sigma=0.1; q=0; nknot=8;
data = SimuEx(n=n, sigma=sigma, q=q, nknot=nknot, seed=0)
res = RunL0L1(data=data, n=n, q=q, TopMethod="top")
ltype = ifelse(q==0, "s", "l")
plot(data$x, data$y, type='p', col='grey', xlab="", ylab="")
lines(data$x, data$y0, col=1, lty=2, lwd=1, type=ltype)
lines(data$x, res$L0fit, col=2, lwd=1, type=ltype)
lines(data$x, res$L1fit, col=4, lwd=1, type=ltype)
points(res$L0knot, rep(par("usr")[4], length(res$L0knot)), pch=3, cex=2)
legend("bottom", c("True", "L0-TF", "L1-TF"), lty=c(2,1,1), col=c(1,2,4),
       horiz=T,  bty="n", pt.lwd=0.2,
       cex=0.8, x.intersp = 0.2, text.width=0.05)
title(main="Blocks Example (Equal-spaced Knots)")


n=300; sigma=0.1; q=0; nknot=8;
data = SimuEx(n=n, sigma=sigma, q=q, nknot=nknot, RandKnot = T, seed=0)
res = RunL0L1(data=data, n=n, q=q, TopMethod="top")
ltype = ifelse(q==0, "s", "l")
plot(data$x, data$y, type='p', col='grey', xlab="", ylab="")
lines(data$x, data$y0, col=1, lty=2, lwd=1, type=ltype)
lines(data$x, res$L0fit, col=2, lwd=1, type=ltype)
lines(data$x, res$L1fit, col=4, lwd=1, type=ltype)
points(res$L0knot, rep(par("usr")[4], length(res$L0knot)), pch=3, cex=2)
legend("bottom", c("True", "L0-TF", "L1-TF"), lty=c(2,1,1), col=c(1,2,4),
       horiz=T,  bty="n", pt.lwd=0.2,
       cex=0.8, x.intersp = 0.2, text.width=0.05)
title(main="Blocks Example (Unequal-spaced Knots)")
```

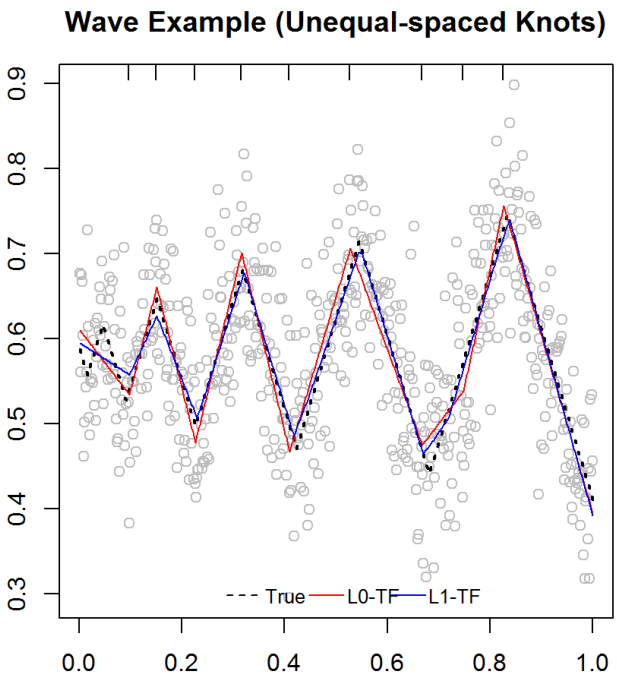




3.2 Piecewise Linear Case

```
# ------------------------------------
# Run Piecewise-linear: L1TF vs. L0TF
# n = 600
# Adaptive knots for Piecewise linear, seed=1
# ------------------------------------
par(mfrow=c(1,2),mar=c(3,3,3,3))
# Piecewise linear case
n=600; sigma=0.1/sqrt(2); q=1; nknot=9; seed=0
data = SimuEx(n=n, sigma=sigma, q=q, nknot=nknot, seed=seed)
res = RunL0L1(data=data, n=n, q=q, TopMethod = "pks")
ltype = ifelse(q==0, "s", "l")
plot(data$x, data$y, type='p', col='grey', xlab="", ylab="")
lines(data$x, data$y0, col=1, lty=3, lwd=2, type=ltype)
lines(data$x, res$L0fit, col=2, lwd=1, type=ltype)
lines(data$x, res$L1fit, col=4, lwd=1, type=ltype)
points(res$L0knot, rep(par("usr")[4], length(res$L0knot)), pch=3, cex=2)
legend("bottom", c("True", "L0-TF", "L1-TF"), lty=c(2,1,1), col=c(1,2,4),
       horiz=T,  bty="n", pt.lwd=0.2,
       cex=0.8, x.intersp = 0.2, text.width=0.05)
title(main="Wave Example (Equal-spaced Knots)")


n=600; sigma=0.1/sqrt(2); q=1; nknot=10; seed=0
data = SimuEx(n=n, sigma=sigma, q=q, nknot=nknot, AdaKnot = T, seed=seed)
res = RunL0L1(data=data, n=n, q=q, TopMethod="top")
ltype = ifelse(q==0, "s", "l")
plot(data$x, data$y, type='p', col='grey', xlab="", ylab="")
lines(data$x, data$y0, col=1, lty=3, lwd=2, type=ltype)
lines(data$x, res$L0fit, col=2, lwd=1, type=ltype)
lines(data$x, res$L1fit, col=4, lwd=1, type=ltype)
points(res$L0knot, rep(par("usr")[4], length(res$L0knot)), pch=3, cex=2)
legend("bottom", c("True", "L0-TF", "L1-TF"), lty=c(2,1,1), col=c(1,2,4),
       horiz=T,  bty="n", pt.lwd=0.2,
       cex=0.8, x.intersp = 0.2, text.width=0.05)
title(main="Wave Example (Unequal-spaced Knots)")
```

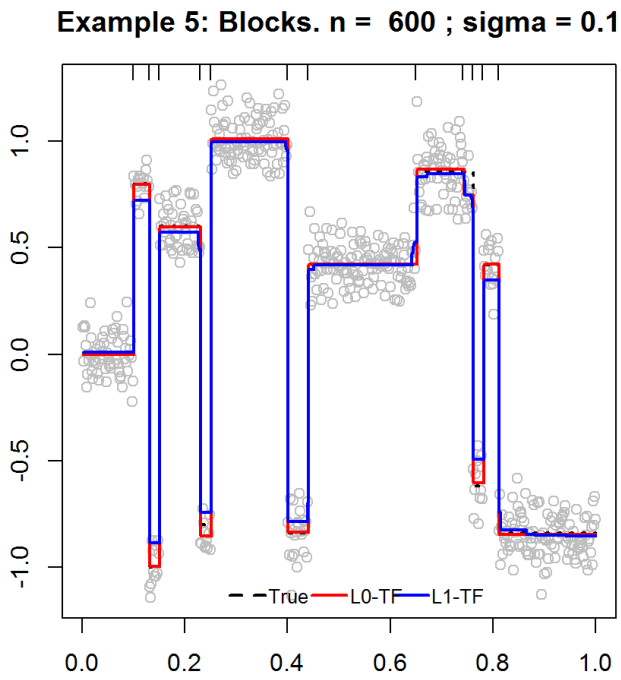


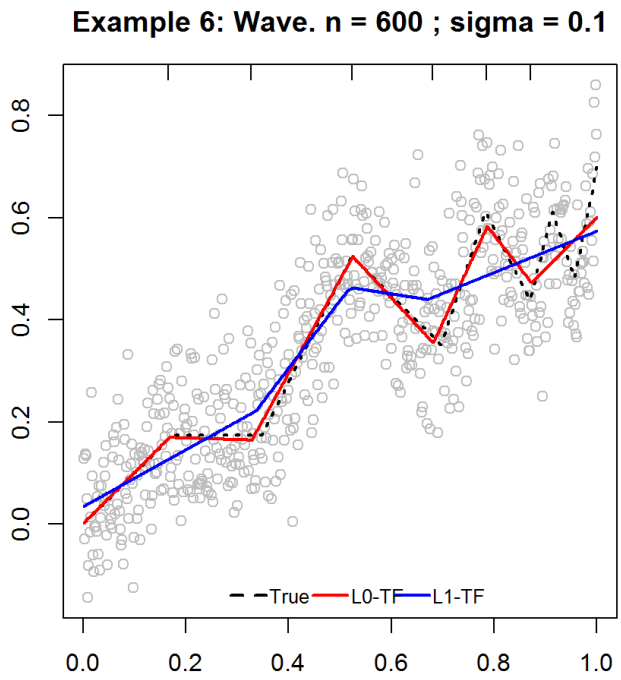


4 Demonstration for the simulation data in L0-TF paper

4.1 Blocks and Wave Simulations

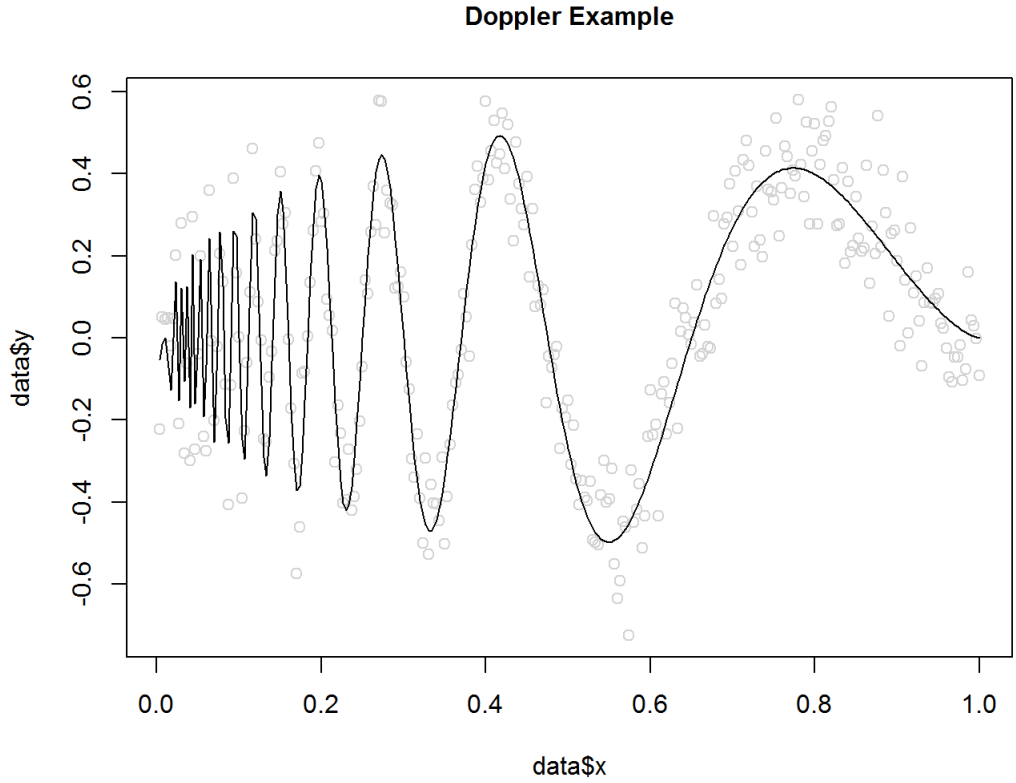
```
par(mfrow=c(1,2),mar=c(3,3,3,3))
library(AMIAS)
n = 600; sigma=0.1

## Blocks example ----
q=0; set.seed(0)
data = SimuBlocks(n, sigma = sigma)
data$x = seq(1/n, 1,length.out = n)
resL0 = samias(as.numeric(data$y), D_type="tf0", kmax=length(data$SetA)+4,tmax=10, adjust = TRUE)
resL1 = L1TF(data, n, q)



ltype = ifelse(q==0, "s", "l")
plot(data$x, data$y, type='p', col='grey', xlab="", ylab="")
lines(data$x, data$y0, col=1, lty=3, lwd=2, type=ltype)
lines(data$x, resL0$alpha, col=2, lwd=2, type=ltype)
lines(data$x, resL1$beta, col=4, lwd=2, type=ltype)
points(resL0$A/n, rep(par("usr")[4], length(resL0$A)), pch=3, cex=2)
legend("bottom", c("True", "L0-TF", "L1-TF"), lty=c(2,1,1), col=c(1,2,4), lwd=2,
       horiz=T,  bty="n", pt.lwd=0.2,
       cex=0.8, x.intersp = 0.2, text.width=0.05)
title(main=paste("Example 5: Blocks. n = ", n, "; sigma =", sigma))

## Wave example ----
q=1; set.seed(0)
data = SimuWave(n, sigma = sigma)
data$x = seq(1/n, 1,length.out = n)
resL0 = samias(as.numeric(data$y), D_type="tfq", q = q, kmax=length(data$SetA)+4,tmax=10, adjust = TRUE)
resL1 = L1TF(data, n, q)

ltype = ifelse(q==0, "s", "l")
plot(data$x, data$y, type='p', col='grey', xlab="", ylab="")
lines(data$x, data$y0, col=1, lty=3, lwd=2, type=ltype)
lines(data$x, resL0$alpha, col=2, lwd=2, type=ltype)
lines(data$x, resL1$beta, col=4, lwd=2, type=ltype)
points(resL0$A/n, rep(par("usr")[4], length(resL0$A)), pch=3, cex=2)
legend("bottom", c("True", "L0-TF", "L1-TF"), lty=c(2,1,1), col=c(1,2,4), lwd=2,
       horiz=T,  bty="n", pt.lwd=0.2,
       cex=0.8, x.intersp = 0.2, text.width=0.05)
title(main=paste("Example 6: Wave. n =", n, "; sigma =", sigma))
```



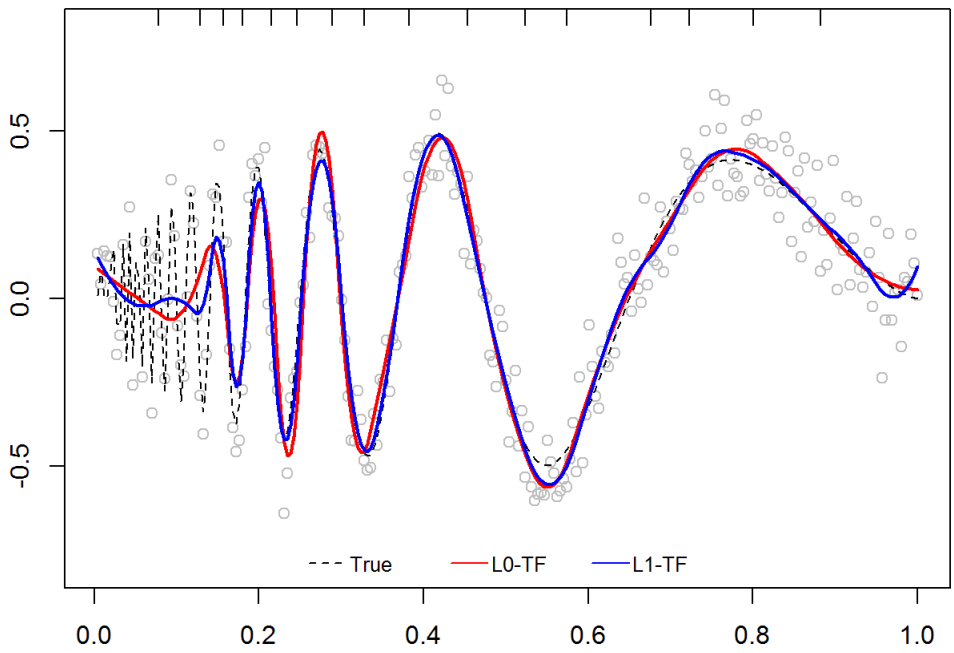


4.2 Doppler Simulations

```
SimuDoppler <- function(n, sigma = 0.1, seed=NA){
  if (!is.na(seed)) set.seed(seed)
  x <- seq(1/n, 1,length.out = n)
  y0 <- sqrt(x*(1-x))*sin(2*pi*(1+0.05)/(x+0.05))
  y <- y0 + sigma*rnorm(n)
  return(list(y = y, y0 = y0, x=x))
}

# Testing simulation
data = SimuDoppler(n=300, sigma=0.1)
par(mfrow=c(1,1))
plot(data$x, data$y, type='p', col='lightgrey')
lines(data$x, data$y0, col="black", lwd=1, type="l")
title(main="Doppler Example", cex.main=1)
```




```
n=256; sigma=0.1; q=2; seed=0
data = SimuDoppler(n=n, sigma=sigma, seed=seed)
maxknot = 29

res = samias(as.numeric(data$y), D_type="tfq", q = q, kmax=maxknot,tmax=10, adjust = TRUE, delta = 2)
resL1 = L1TF(data, n, q)

par(mfrow=c(1,1))
ltype = ifelse(q==0, "s", "l")
plot(data$x, data$y, type='p', col='grey', xlab="", ylab="", ylim=c(-0.8, 0.8))
lines(data$x, data$y0, col=1, lty=2, lwd=1, type=ltype)
lines(data$x, res$alpha, col=2, lwd=2, type=ltype)
lines(data$x, resL1$beta, col=4, lwd=2, type=ltype)
points(res$A/n, rep(par("usr")[4], length(res$A)), pch=3, cex=2)
legend("bottom", c("True", "L0-TF", "L1-TF"), lty=c(2,1,1), col=c(1,2,4),
       horiz=T,  bty="n", pt.lwd=0.2,
       cex=0.8, x.intersp = 0.2, text.width=0.1)
title(main=paste("Doppler Example (Inhomogeneous Smoothness): n =", n))
```


Doppler Example (Inhomogeneous Smoothness): n = 256
0.5
0.0
-0.5
True
L0-TF
L1-TF
0.0
0.2
0.4
0.6
0.8
1.0