Simulations

Hadrien Lorenzo

2023-10-06

Load package SEPaLS and simulation functions

```
library(SEPaLS)
source("../R/functions.R")
```

```
##
## Attachement du package : 'dplyr'
```

```
## Les objets suivants sont masqués depuis 'package:stats':
##
##     filter, lag
```

```
## Les objets suivants sont masqués depuis 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
##
## Attachement du package : 'data.table'
```

```
## Les objets suivants sont masqués depuis 'package:dplyr':
##
##     between, first, last
```

```
##
## Attachement du package : 'VineCopula'
```

```
## L'objet suivant est masqué depuis 'package:copula':
##
##     pobs
```

```
load("../data/data_statists.RData") ## Load saved results
```

Set workspace

```
## Dimension & sample size
# Two dimension
p1 <- 3
p2 <- 30
# Selected dimension : p1 or p2
p <- p2
# Sample size
n <- 500

## Beta
```

```r
beta <- c(rep(1,2),rep(0,p-2))/sqrt(2)

## Link function power
c_s<-c(1,0.5,0.25)

## Noise
# Standard deviation of epsilon
sigma <- 0.9*diag(p)
# Noise signal ratio
r <- 5

## Pareto parameters for Y distribution
# pareto.params[1] : Pareto index and also equal to 1/gamma where gamma is the tail index
# pareto.params[2] :  minimum possible value of Y
pareto.params <- c(5,2)

## Selected distribution : "pareto" or "student"
dist <- "pareto"

## copula. fam   :
#    0= Independent copula
#    1 = Gaussian copula
#    5 = Frank copula
#    3 = Clayton copula
copula.fam <- 3

## Number of replications
N<-1000

## Kendalls tau parameters parameter
thetas <- c(8,1/2,1/2,8)
signe_taus <- c(-1,-1,1,1)
taus <- c(-0.8,-0.2,0.2,0.8)

## Sparse regularization parameters
lambda <- c(0,1e-4,5e-4,1e-3)
n_lambdas <- length(lambda)

## vMF regularization parameters
kappa0 <- c(0,1e-4,3e-3,1e-2)
n_kappas <- length(kappa0)
mu0_1 <- beta
N_prior_NON_Null <- 15
mu0_2 <- c(rep(1,N_prior_NON_Null),rep(0,p-N_prior_NON_Null))/sqrt(N_prior_NON_Null)


# The 200 largest observations
k.threshold<- n - 100

# Visualization parameters
colors <- viridis::viridis(4)
alpha_transpa <- 0.15
```

Simulations for von Mises-Fisher prior

Start simulations

```
results_1 = results_2 <- list()
for(i_t in 1:length(thetas)){
  set.seed(i_t)
  results_1[[i_t]] <- simu_process(mu0 = mu0_1,kappa0 = kappa0,n,p,beta,
                                   c_s,sigma,r,dist,
                                   pareto.params,
                                   copula.fam = copula.fam,
                                   copula.param = thetas[i_t],
                                   N,
                                   k.threshold,type="vMF",
                                   signe_tau = signe_taus[i_t])
  results_2[[i_t]] <- simu_process(mu0 = mu0_2,kappa0 = kappa0,n,p,beta,
                                   c_s,sigma,r,dist,
                                   pareto.params,
                                   copula.fam = copula.fam,
                                   copula.param = thetas[i_t],
                                   N,
                                   k.threshold,type="vMF",
                                   signe_tau = signe_taus[i_t])
}
```

Observe results

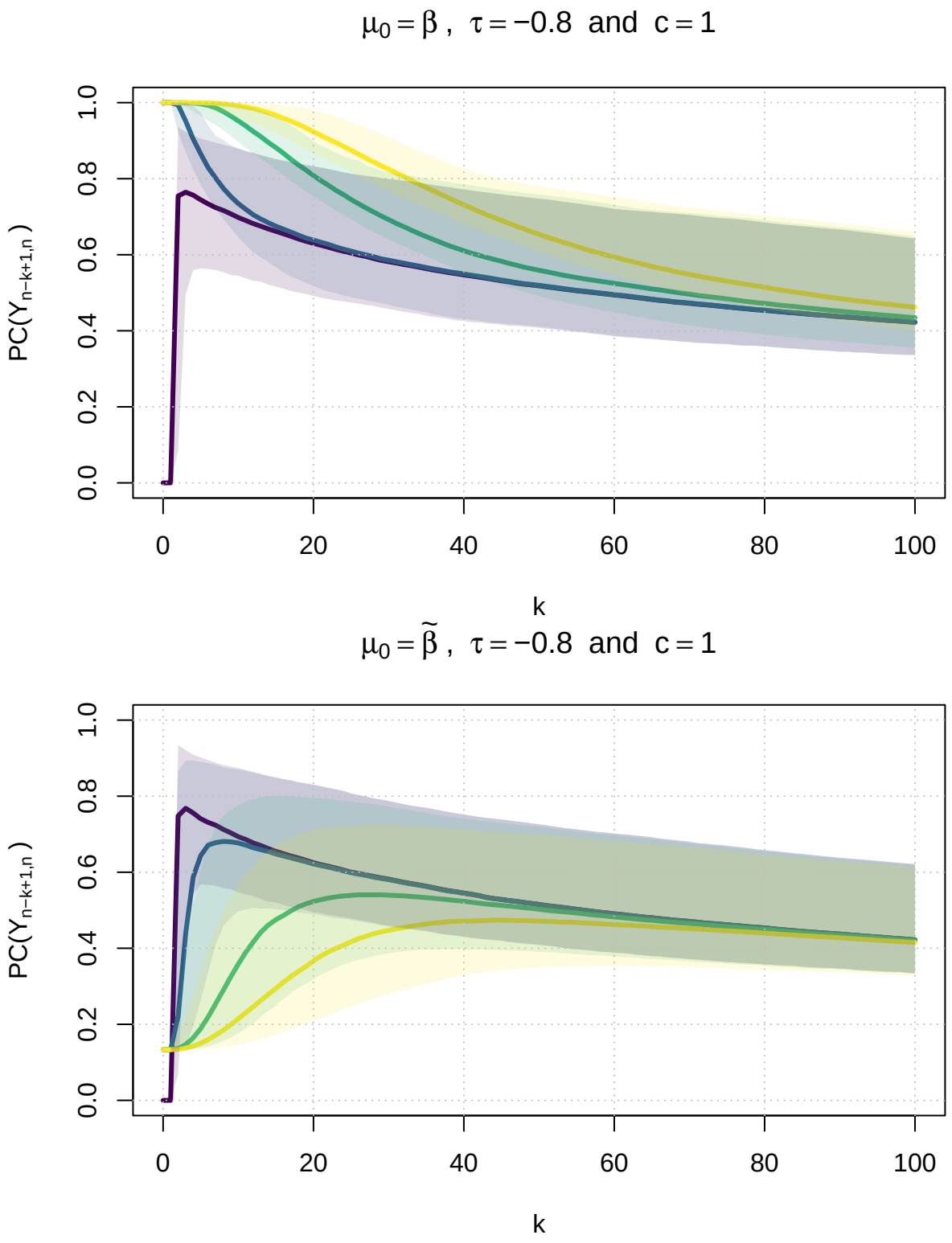
Use the following function

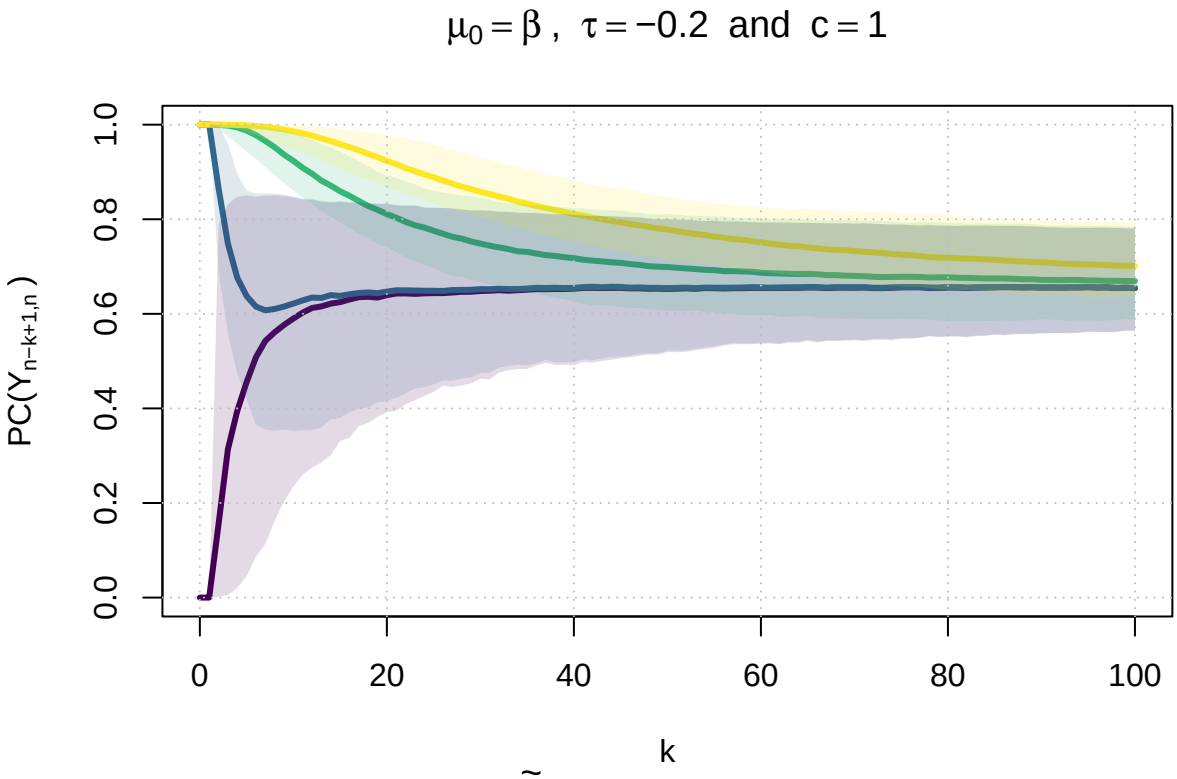
```
observe_vMF <- function(j){
  for(i_t in 1:length(thetas)){
    cos.rep_1 <- data.frame(results_1[[i_t]])
    cos.rep_2 <- data.frame(results_2[[i_t]])
    oo_beta_1=oo_beta_2 <- NULL
    file_1 <- paste("../SEPaLS_simulations/results/vMF/simu_vMF_c",j,
                    "_Theta",i_t,
                    "_mu1.pdf",sep="")
    file_2 <- paste("../SEPaLS_simulations/results/vMF/simu_vMF_c",j,
                    "_Theta",i_t,
                    "_mu2.pdf",sep="")
    for(i in 1:n_lambdas){
      id <- which(cos.rep_1$kappa0==i &
                    cos.rep_1$id_power==j)
      if(i==1){
        oo_beta_1_median <- cos.rep_1$median[id]
        oo_beta_2_median <- cos.rep_2$median[id]
        oo_beta_1_quant5 <- cos.rep_1$quant5[id]
        oo_beta_2_quant5 <- cos.rep_2$quant5[id]
        oo_beta_1_quant95 <- cos.rep_1$quant95[id]
        oo_beta_2_quant95 <- cos.rep_2$quant95[id]
      }
      else
      {
        oo_beta_1_median <- cbind(oo_beta_1_median,
```

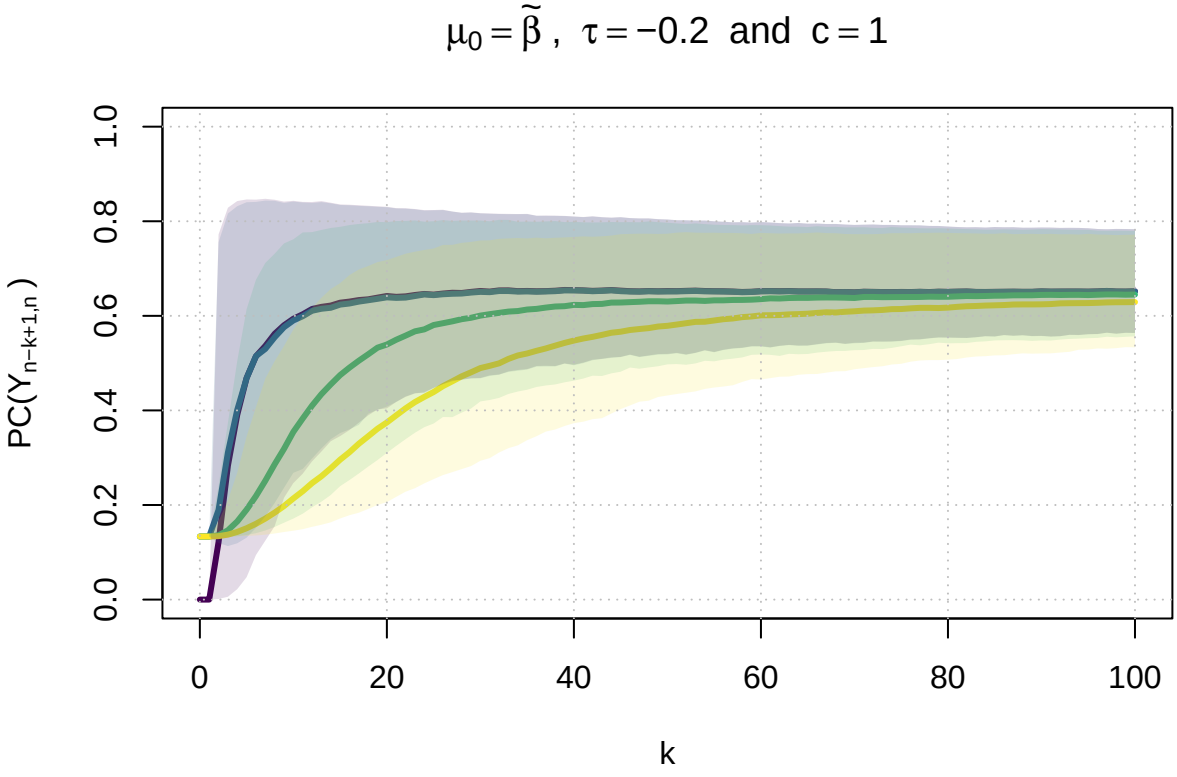
```
                                cos.rep_1$median[id])
      oo_beta_2_median <- cbind(oo_beta_2_median,
                                cos.rep_2$median[id])
      oo_beta_1_quant5 <- cbind(oo_beta_1_quant5,
                                cos.rep_1$quant5[id])
      oo_beta_2_quant5 <- cbind(oo_beta_2_quant5,
                                cos.rep_2$quant5[id])
      oo_beta_1_quant95 <- cbind(oo_beta_1_quant95,
                                cos.rep_1$quant95[id])
      oo_beta_2_quant95 <- cbind(oo_beta_2_quant95,
                                cos.rep_2$quant95[id])
    }
  }
  # pdf(file_1,width = 7,height = 5,onefile = T)
  main <- bquote(mu[0]==beta~", "~tau==.(taus[i_t])~" and "~c==.(c_s[j]))
  XX <- n-unique(cos.rep_1$nb_exceed)
  matplot(XX,oo_beta_1_median,
          type="l",lty=1,lwd=3,main=main,
          ylab=bquote("PC(Y"["n-k+1,n"]~")"),
          xlab=expression(k),
          ylim=c(0,1),
          col=colors)
  for(jj in 1:ncol(oo_beta_1_quant5)){
    polygon(x=c(XX,rev(XX)),
            y=c(oo_beta_1_quant5[,jj],rev(oo_beta_1_quant95[,jj])),
            col=scales::alpha(colors[jj],alpha_transpa),border=NA)
  }
  abline(h=(0:5)/5,col="gray",lty=3)
  abline(v=(0:10)*20,col="gray",lty=3)
  # dev.off()
  # pdf(file_2,width = 7,height = 5,onefile = T)
  main <- bquote(mu[0]==tilde(beta)~", "~tau==.(taus[i_t])~" and "~c==.(c_s[j]))
  matplot(n-unique(cos.rep_2$nb_exceed),oo_beta_2_median,
          type="l",lty=1,lwd=3,main=main,
          ylab=bquote("PC(Y"["n-k+1,n"]~")"),
          xlab=expression(k),
          ylim=c(0,1),
          col=colors)
  for(jj in 1:ncol(oo_beta_2_quant5)){
    polygon(x=c(XX,rev(XX)),
            y=c(oo_beta_2_quant5[,jj],rev(oo_beta_2_quant95[,jj])),
            col=scales::alpha(colors[jj],alpha_transpa),border=NA)
  }
  abline(h=(0:5)/5,col="gray",lty=3)
  abline(v=(0:10)*20,col="gray",lty=3)
  # dev.off()
 }
}
```

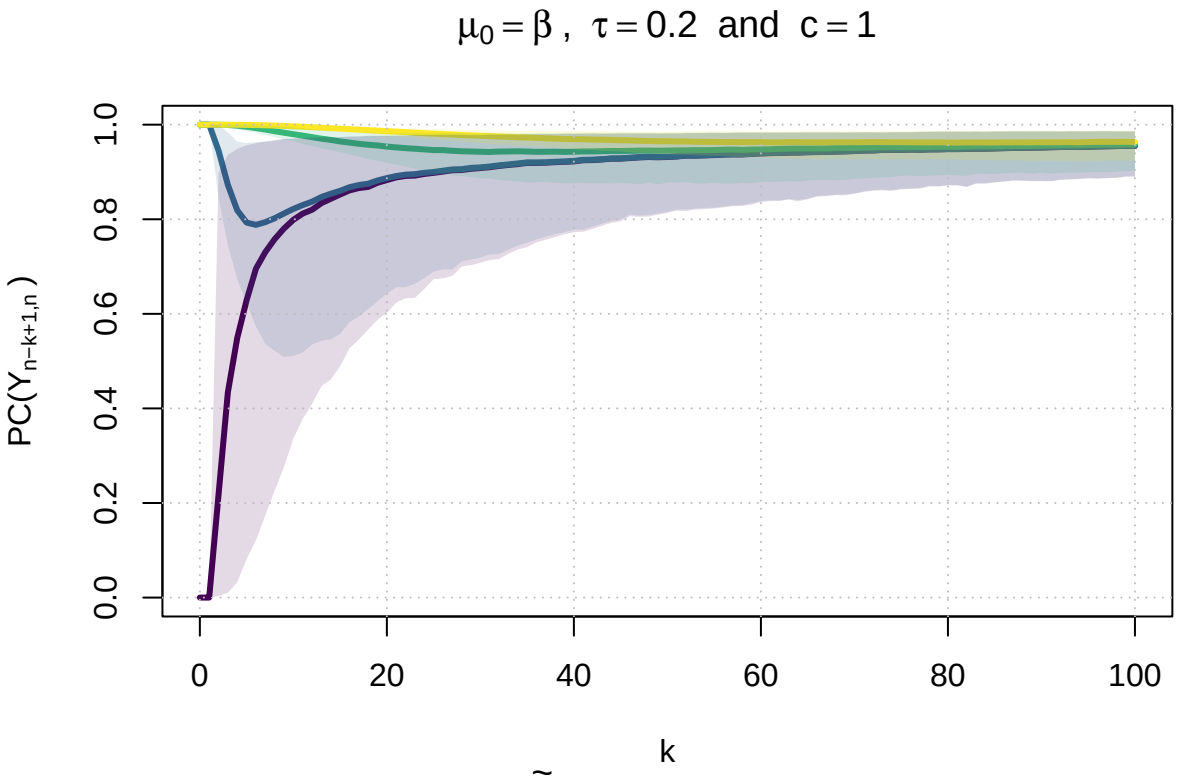
For $c = 1$

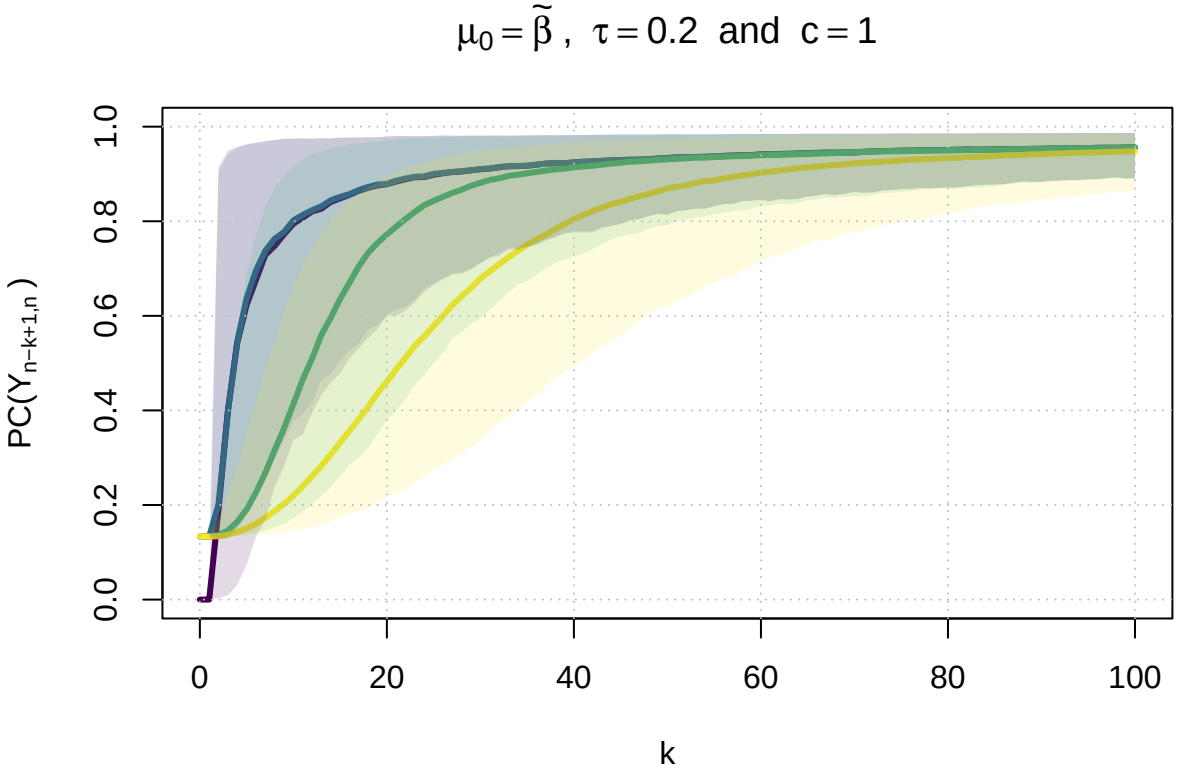
```
observe_vMF(which(c_s==1))
```

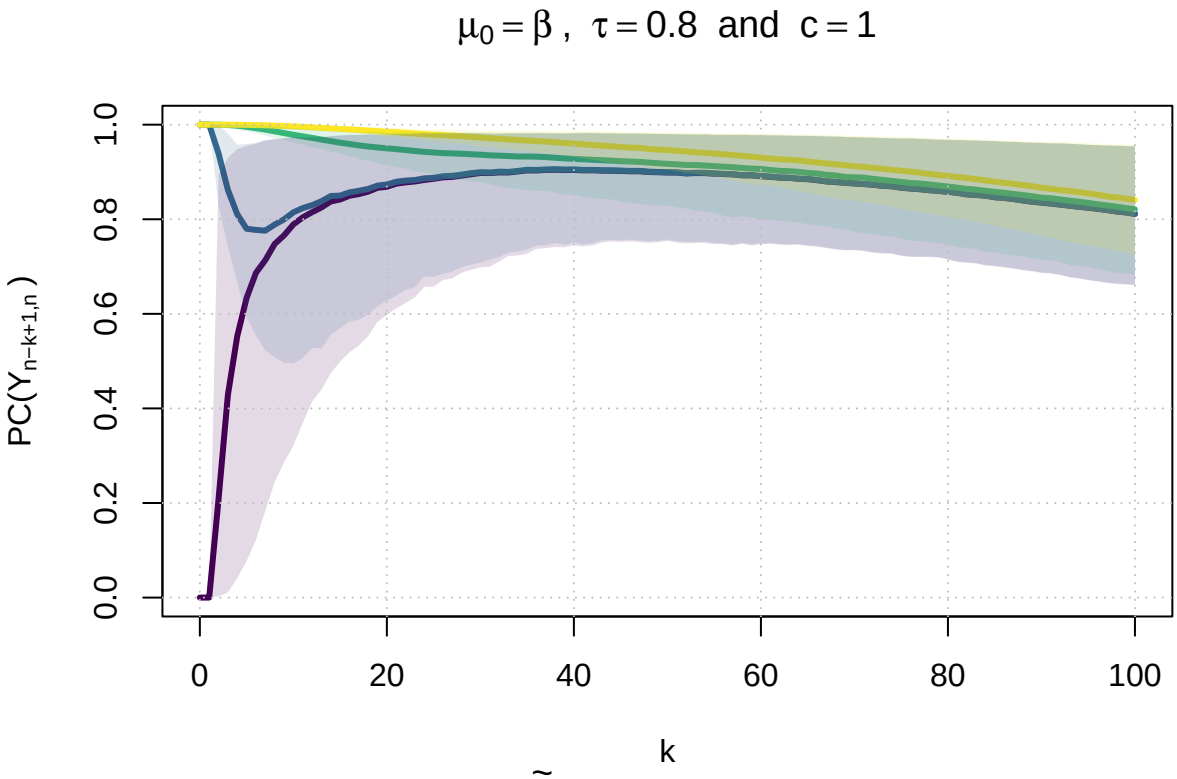


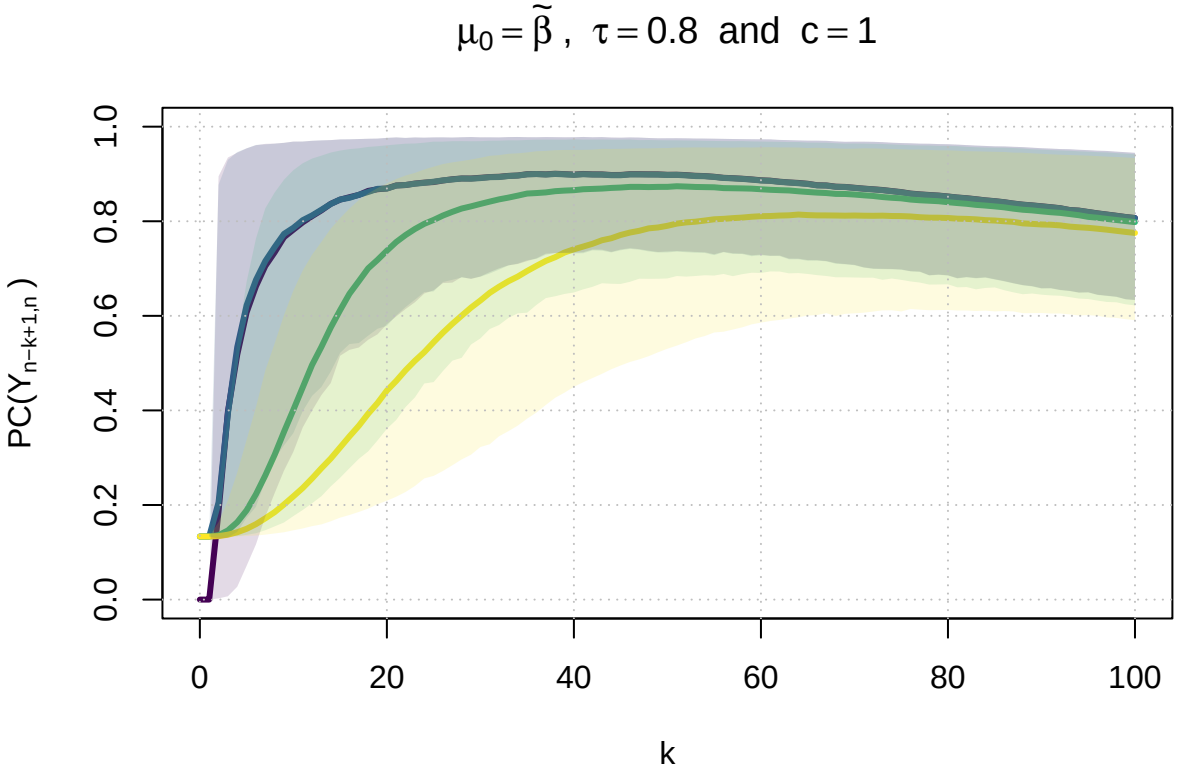

$\mu_0 = \beta$, $\tau = -0.2$ and $c = 1$
$PC(Y_{n-k+1,n})$
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
k


$\mu_0 = \tilde{\beta}$, $\tau = -0.2$ and $c = 1$
$PC(Y_{n-k+1,n})$
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
k


$\mu_0 = \beta$, $\tau = 0.2$ and $c = 1$
PC($Y_{n-k+1,n}$)
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
k

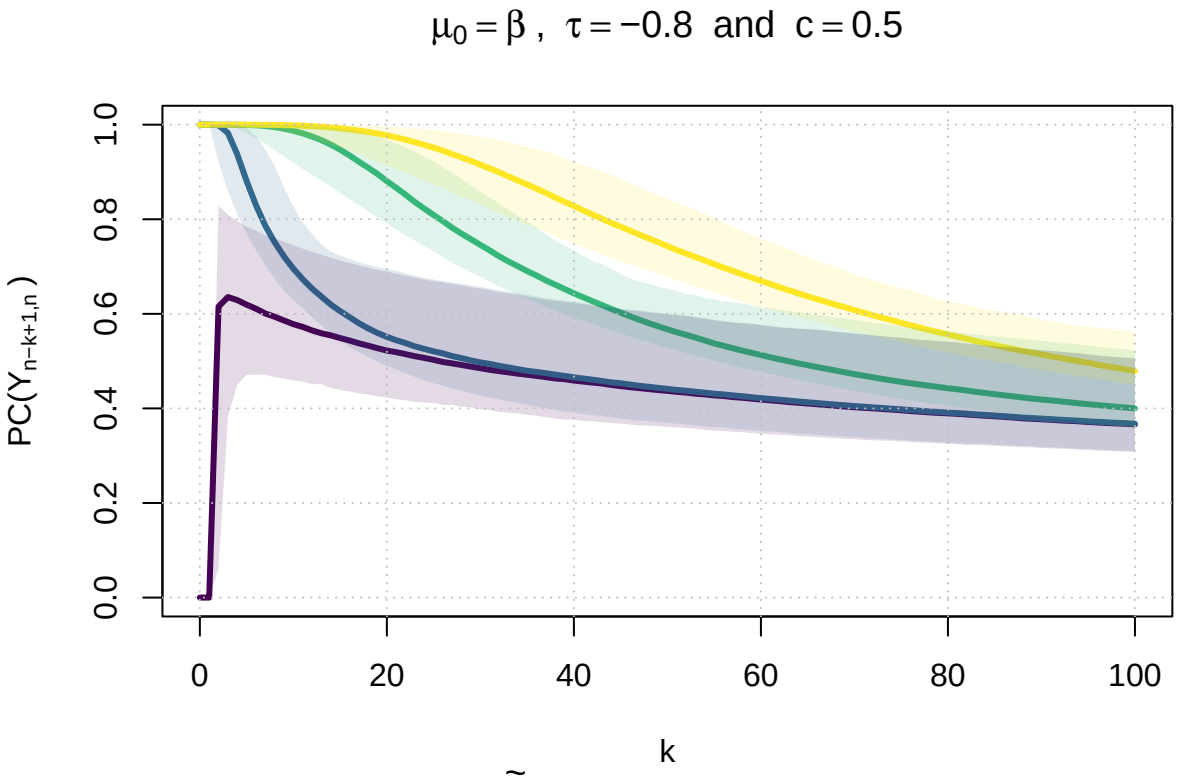

$\mu_0 = \tilde{\beta}$, $\tau = 0.2$ and $c = 1$
PC($Y_{n-k+1,n}$)
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
k

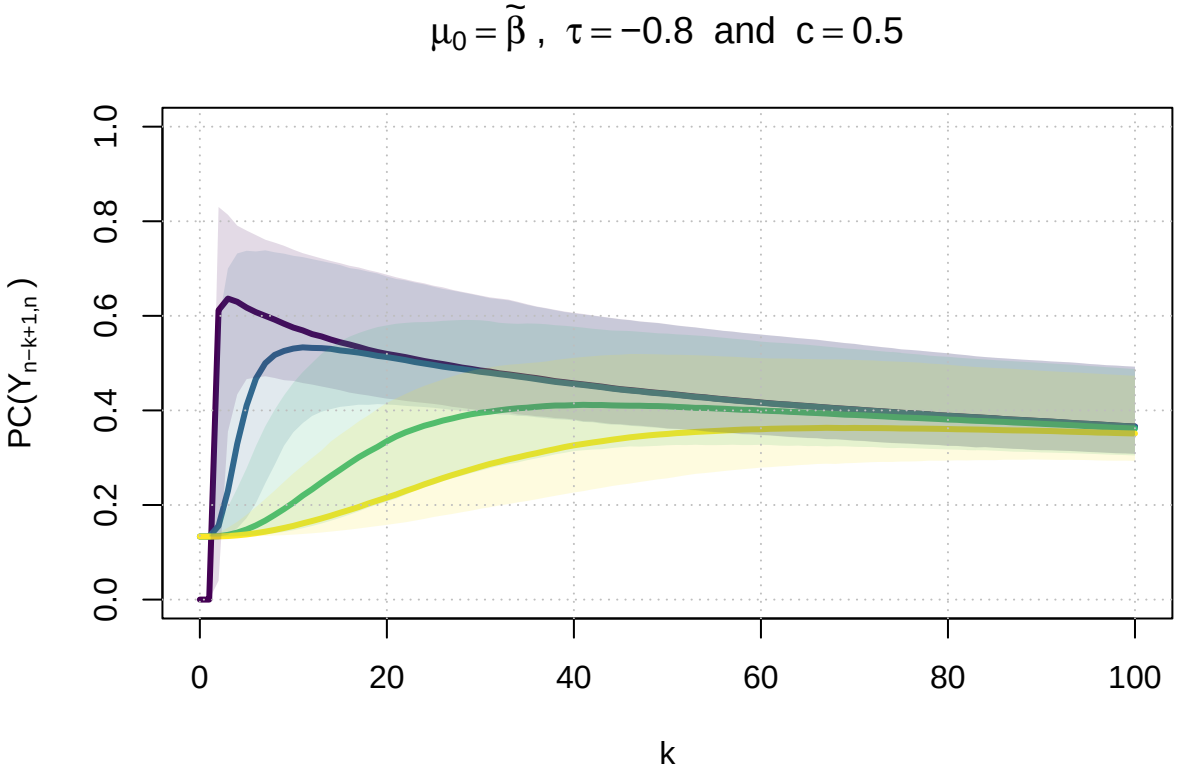


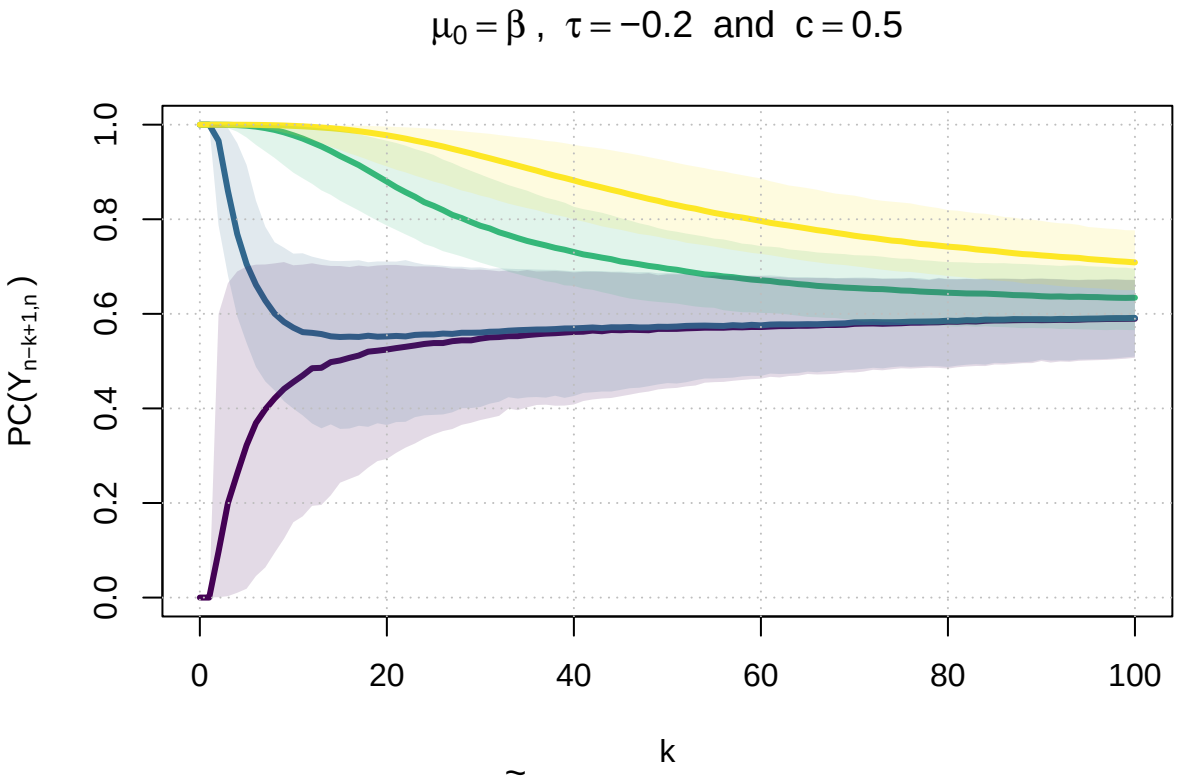


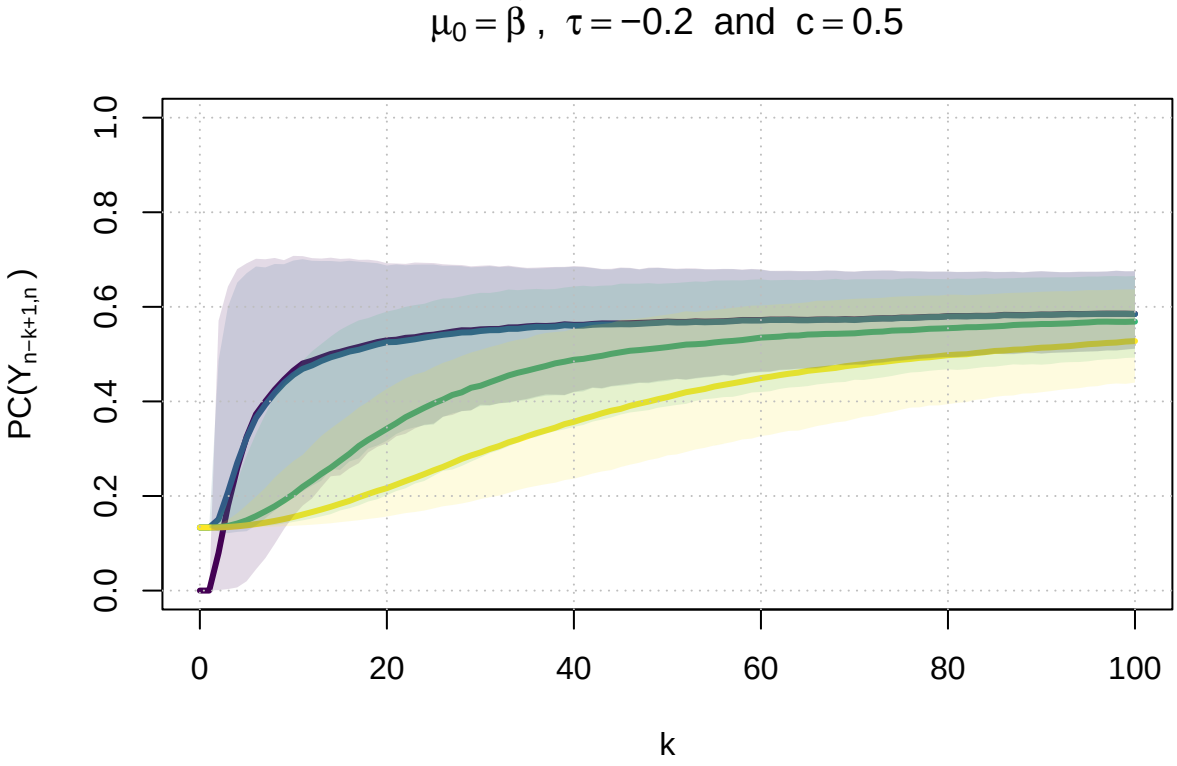
For $c = 1/2$

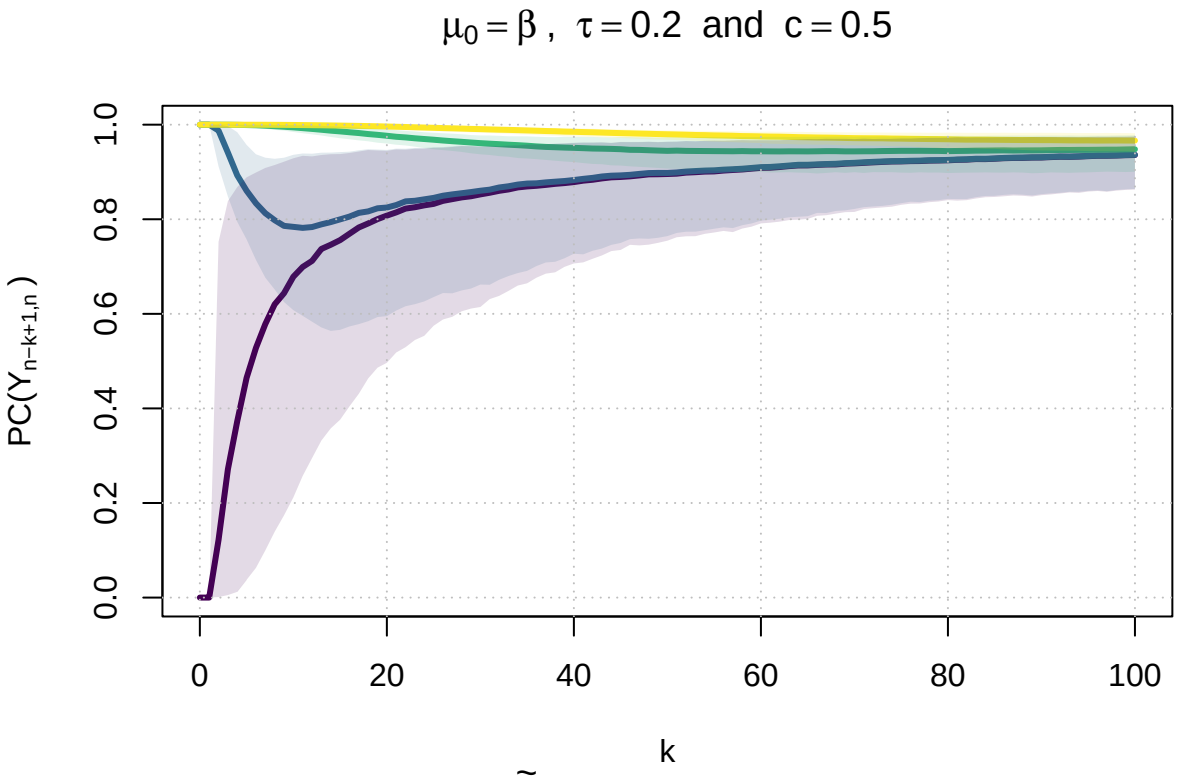
```
observe_vMF(which(c_s==1/2))
```

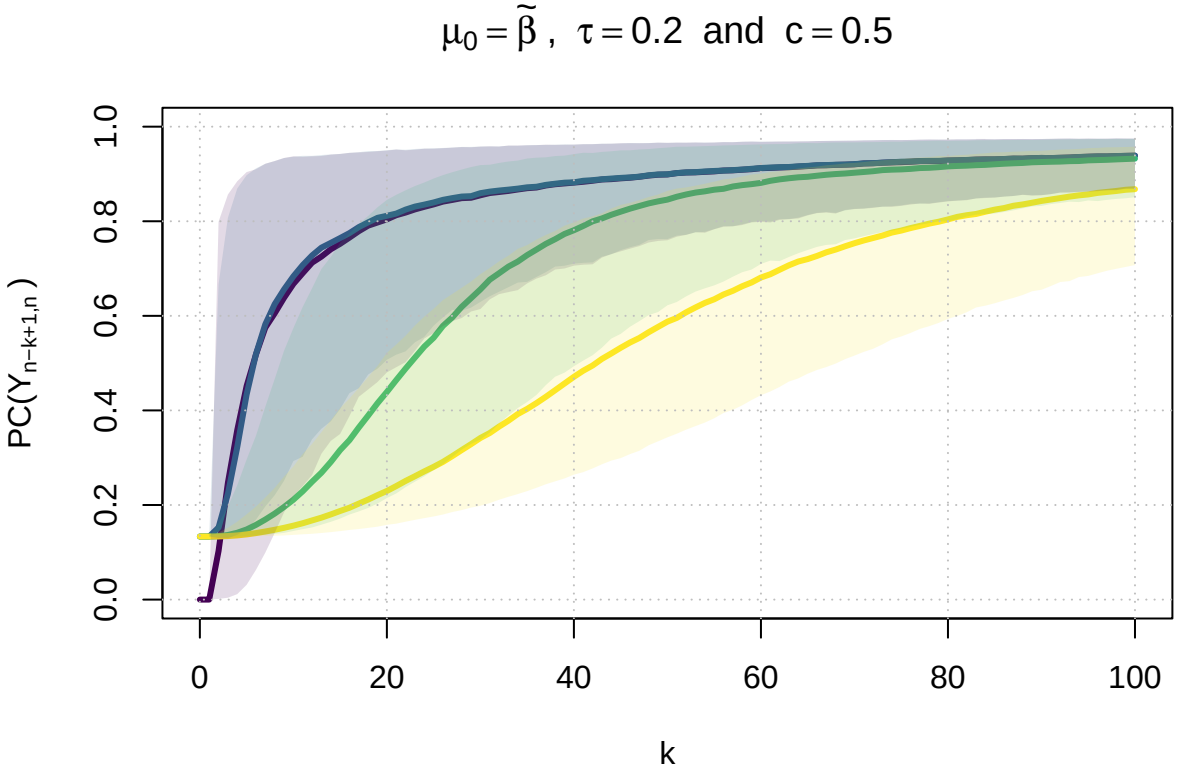

$\mu_0 = \beta$, $\tau = -0.8$ and $c = 0.5$
PC($Y_{n-k+1,n}$)
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
k

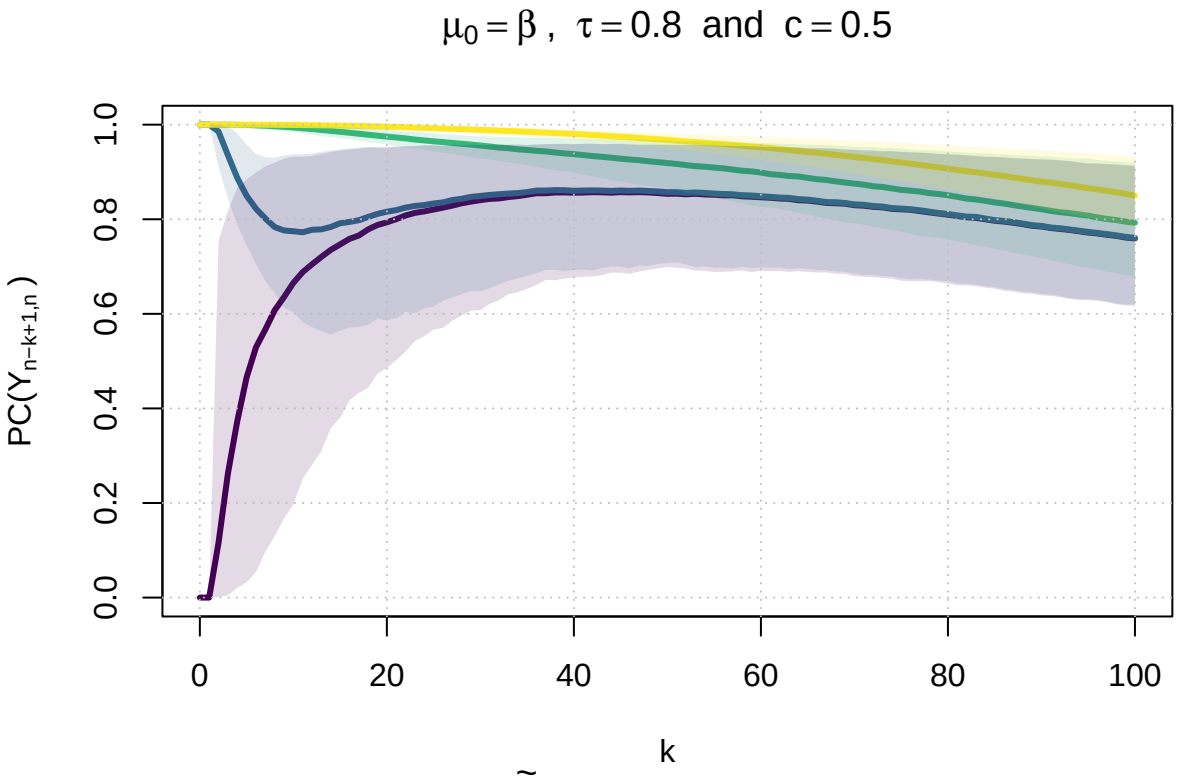

$\mu_0 = \tilde{\beta}$, $\tau = -0.8$ and $c = 0.5$
PC($Y_{n-k+1,n}$)
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
k

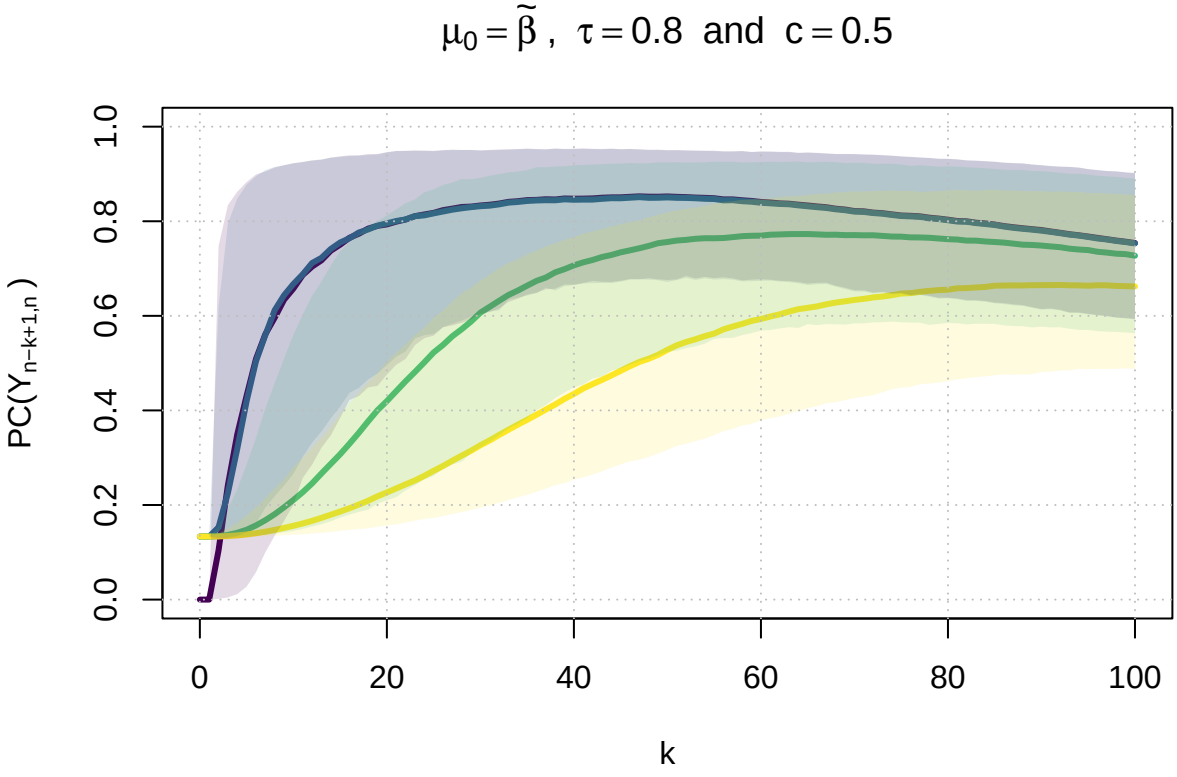

$\mu_0 = \beta$, $\tau = -0.2$ and $c = 0.5$
PC($Y_{n-k+1,n}$)
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
k


$\mu_0 = \tilde{\beta}$, $\tau = -0.2$ and $c = 0.5$
PC($Y_{n-k+1,n}$)
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
k


$\mu_0 = \beta$, $\tau = 0.2$ and $c = 0.5$
$PC(Y_{n-k+1,n})$
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
k

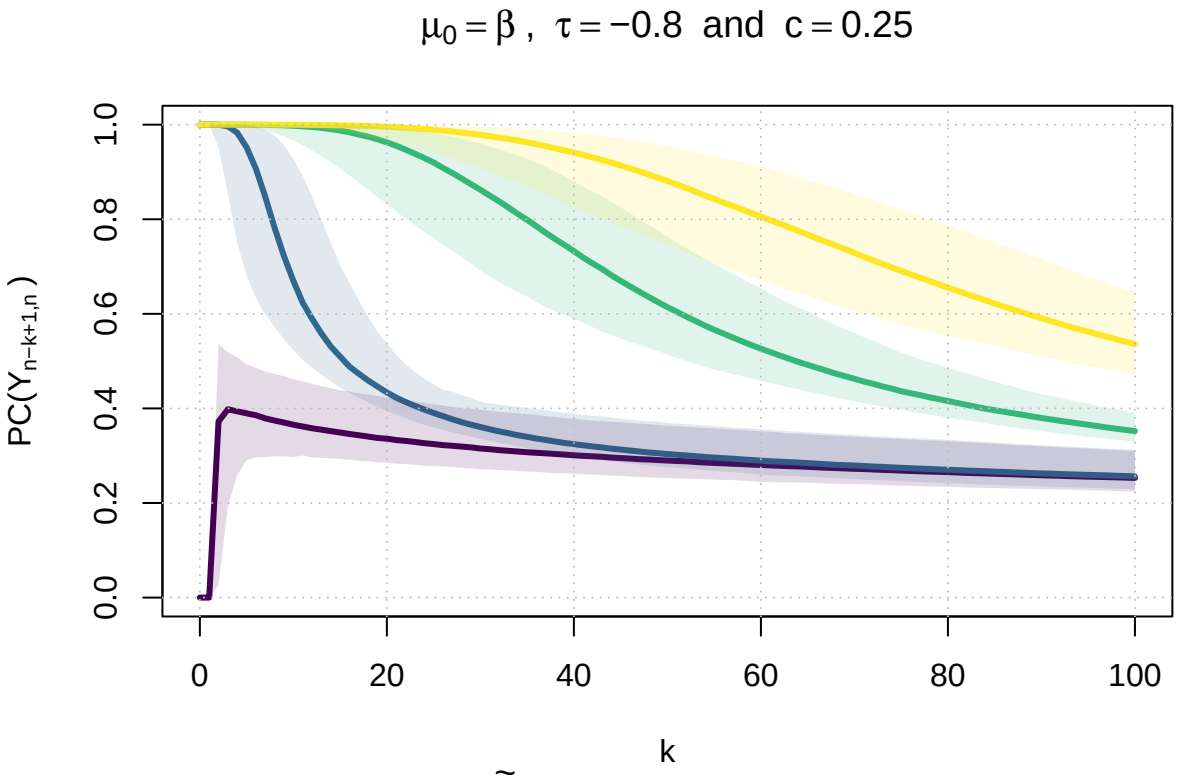

$\mu_0 = \tilde{\beta}$, $\tau = 0.2$ and $c = 0.5$
$PC(Y_{n-k+1,n})$
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
k

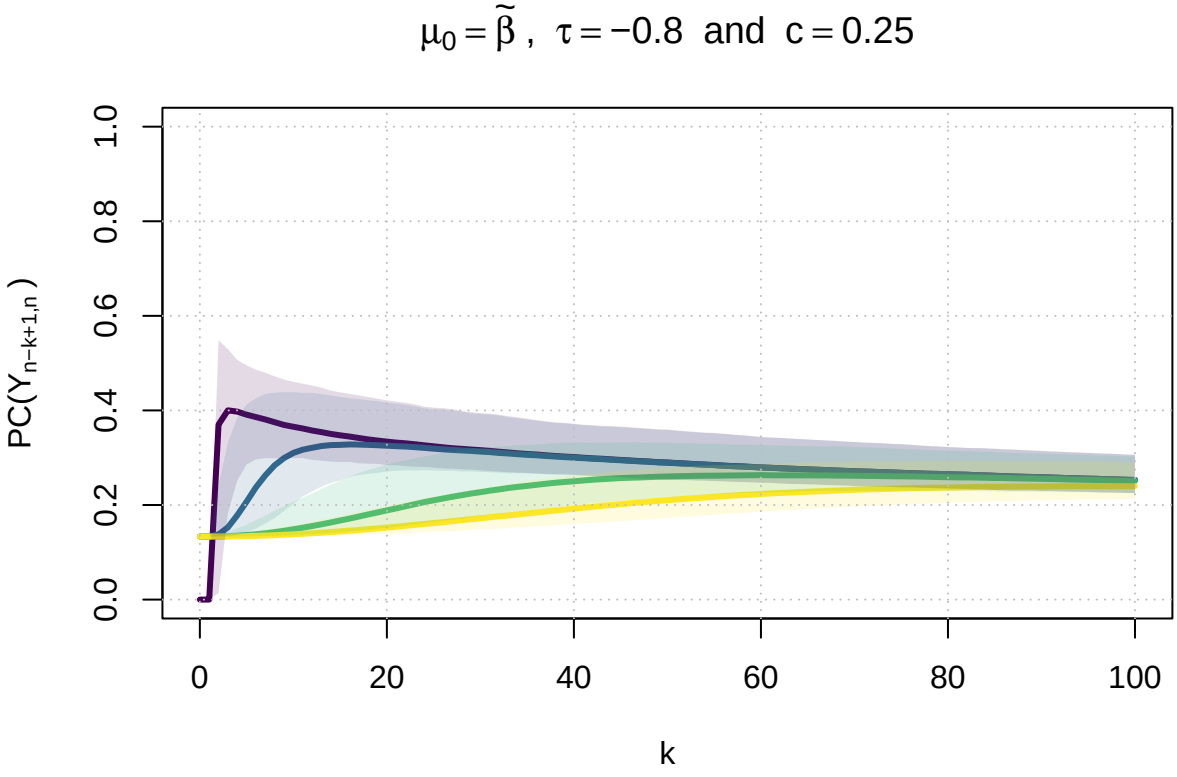


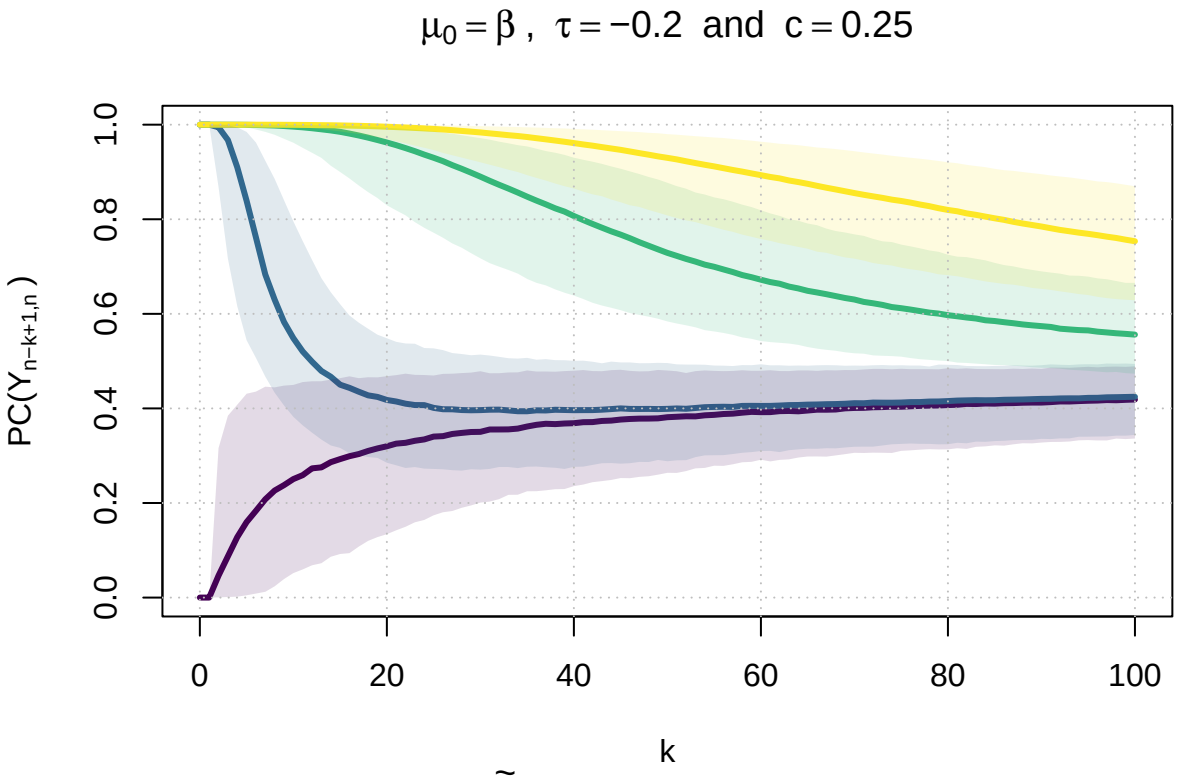


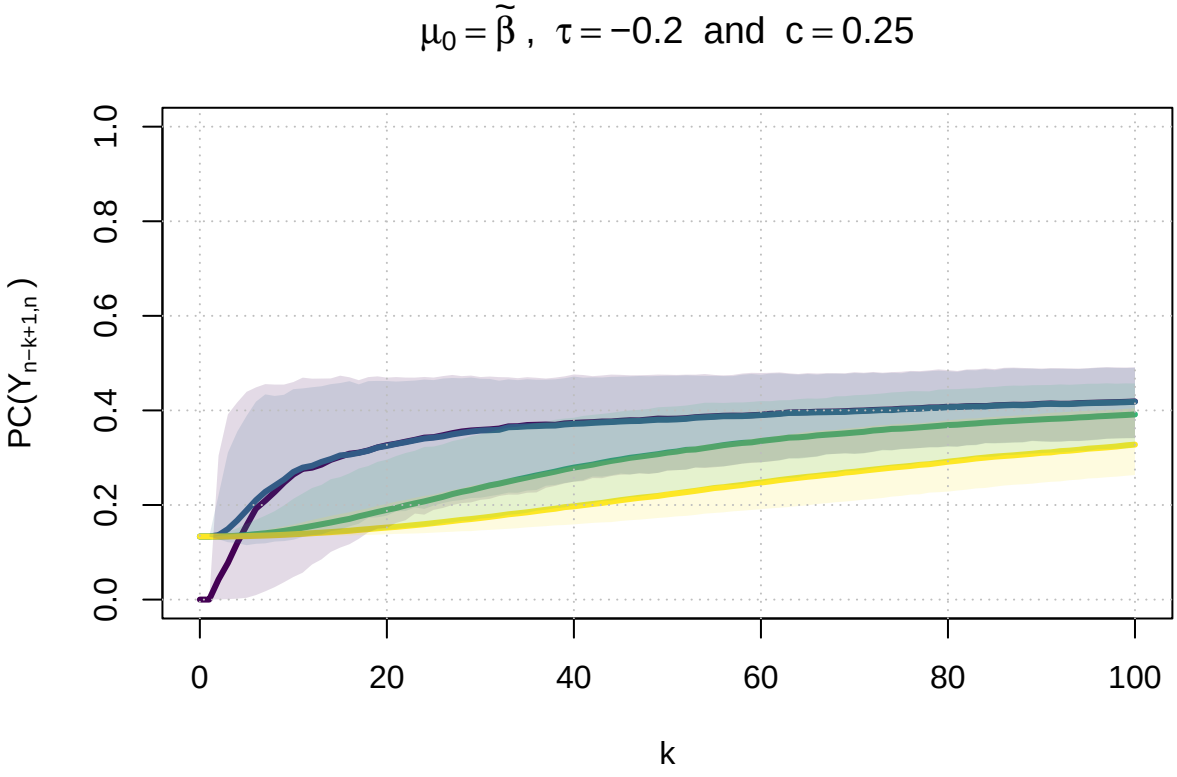
For $c = 1/4$

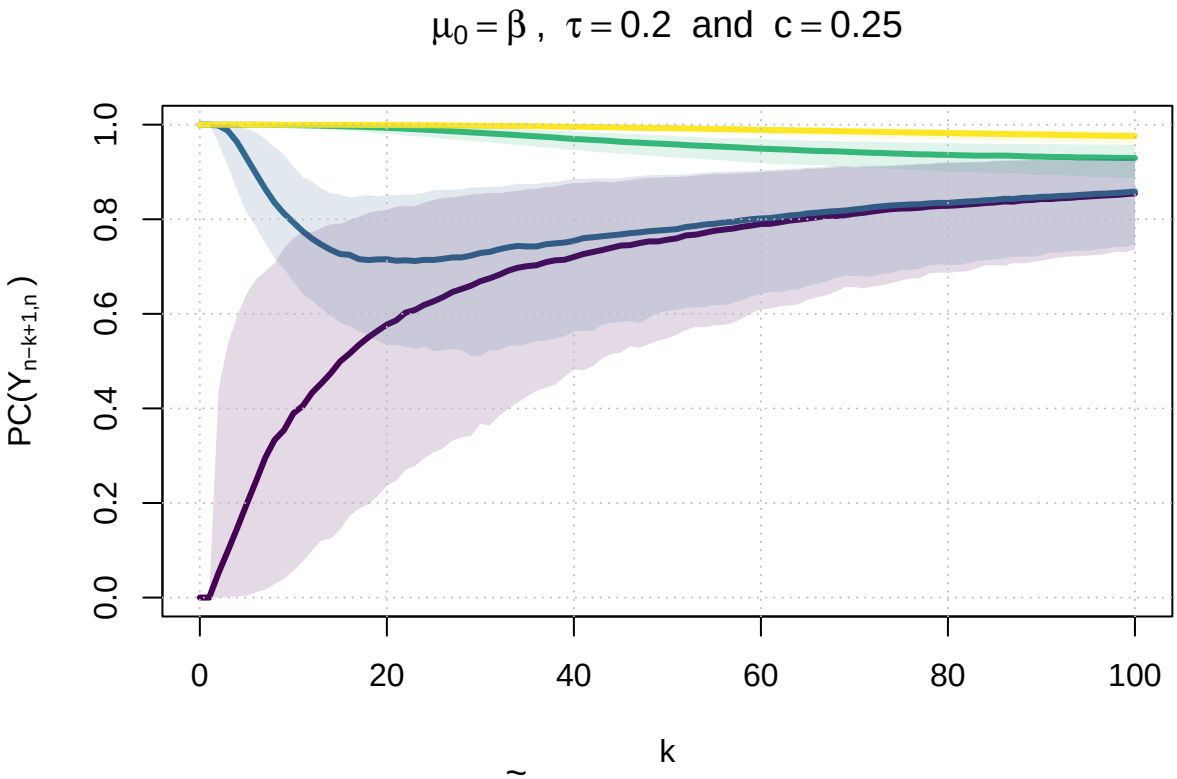
```
observe_vMF(which(c_s==1/4))
```

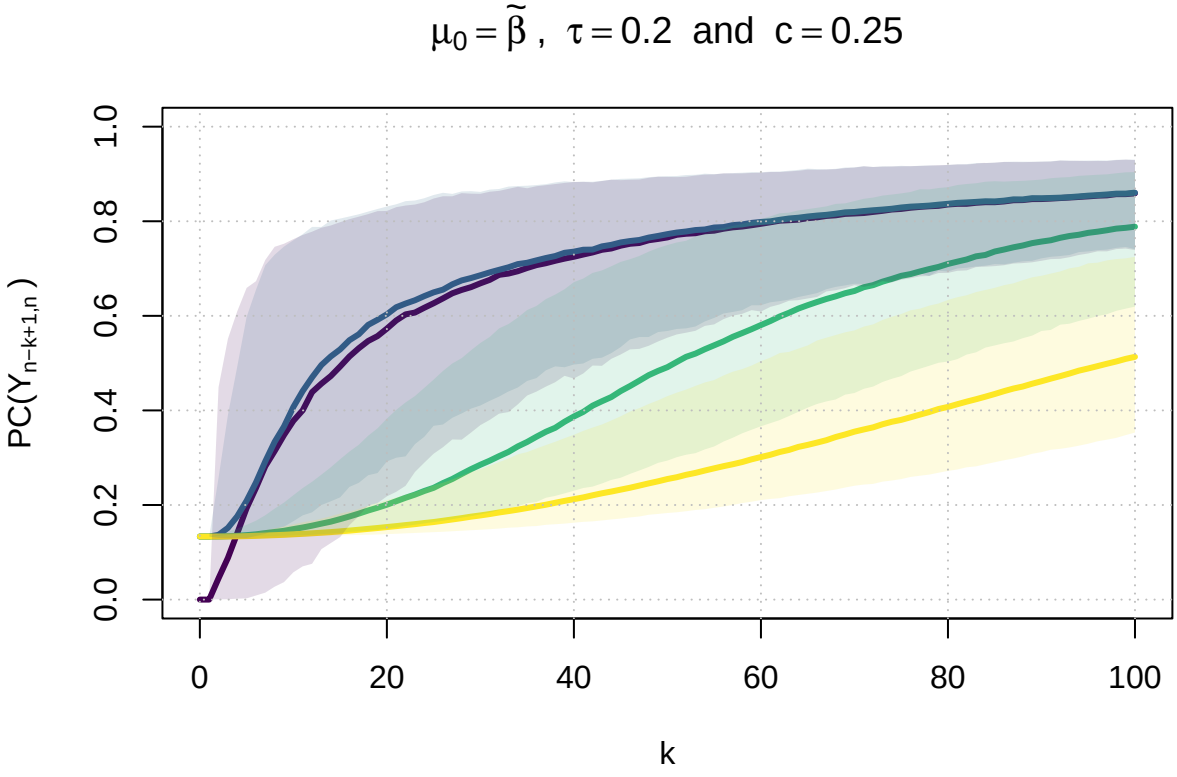

$\mu_0 = \beta$, $\tau = -0.8$ and $c = 0.25$
$PC(Y_{n-k+1,n})$
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
k

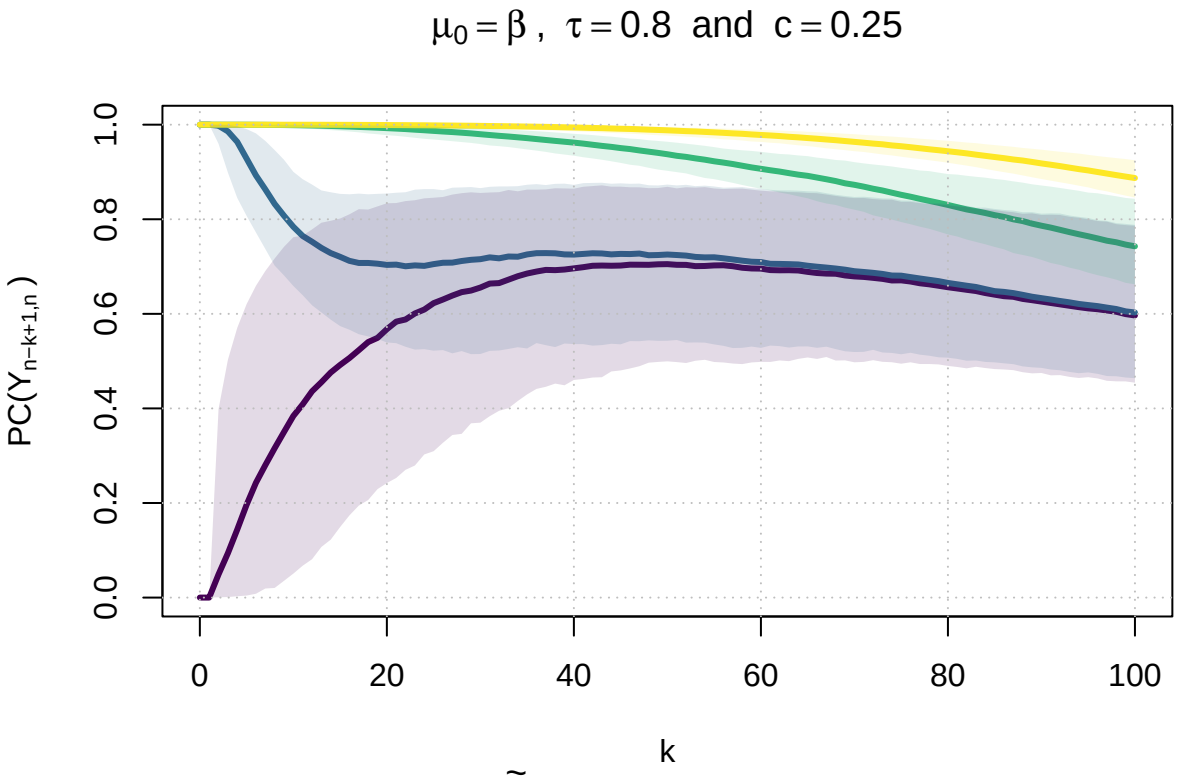

$\mu_0 = \tilde{\beta}$, $\tau = -0.8$ and $c = 0.25$
$PC(Y_{n-k+1,n})$
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
k

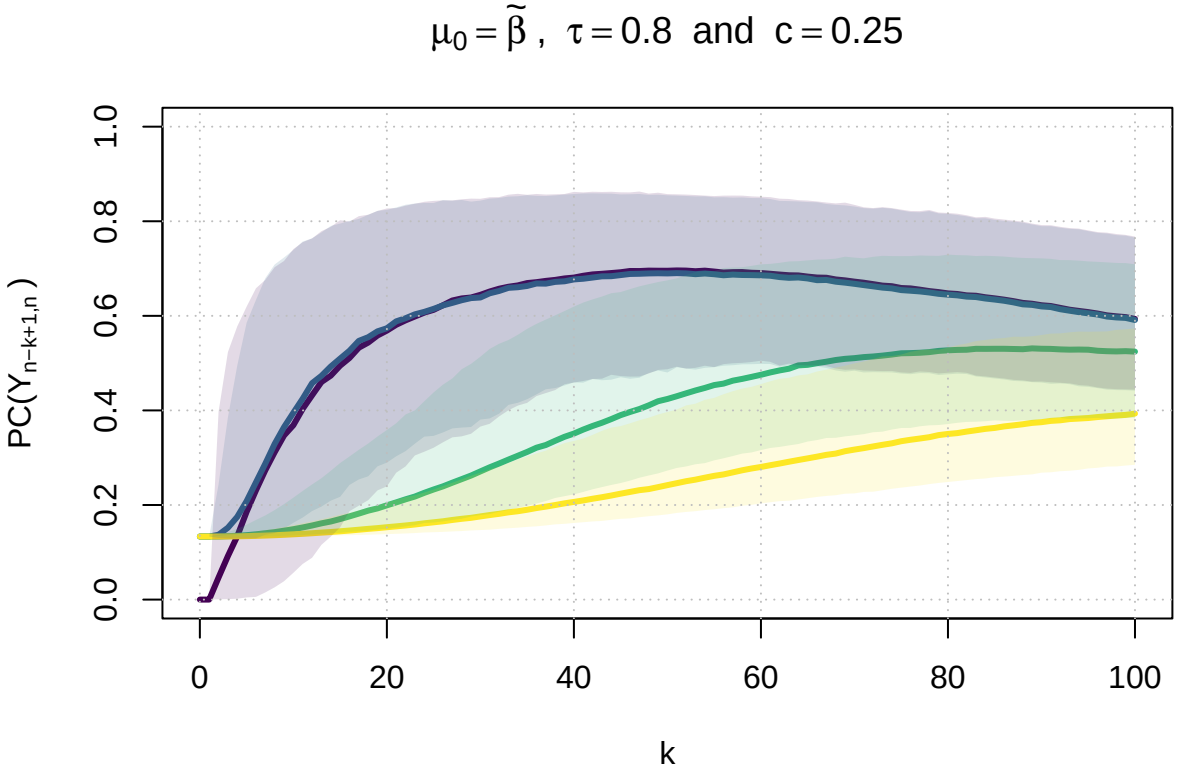

$\mu_0 = \beta$, $\tau = -0.2$ and $c = 0.25$
$PC(Y_{n-k+1,n})$
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
k


$\mu_0 = \tilde{\beta}$, $\tau = -0.2$ and $c = 0.25$
$PC(Y_{n-k+1,n})$
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
k


$\mu_0 = \beta$, $\tau = 0.2$ and $c = 0.25$
$PC(Y_{n-k+1,n})$
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
k


$\mu_0 = \tilde{\beta}$, $\tau = 0.2$ and $c = 0.25$
$PC(Y_{n-k+1,n})$
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
k





Simulations for Sparse prior

```
p_1 <- 30
p_2 <- 300
```

```
beta_30 <- c(rep(1,2),rep(0,p_1-2))/sqrt(2)
beta_300 <- c(rep(1,2),rep(0,p_2-2))/sqrt(2)
```

Start simulations

```
results_sparse_p_1 = results_sparse_p_2 <- list()
for(i_t in 1:length(thetas)){
  set.seed(i_t)
  results_sparse_p_1[[i_t]] <- simu_process(mu0 = mu0_1,kappa0 = lambda,n,p_1,beta_30,
                                    c_s,sigma,r,dist,
                                    pareto.params,
                                    copula.fam = copula.fam,
                                    copula.param = thetas[i_t],
                                    N,
                                    k.threshold,type="Laplace",
                                    signe_tau = signe_taus[i_t])
  results_sparse_p_2[[i_t]] <- simu_process(mu0 = mu0_2,kappa0 = lambda,n,p_2,beta_300,
                                    c_s,sigma,r,dist,
                                    pareto.params,
                                    copula.fam = copula.fam,
                                    copula.param = thetas[i_t],
                                    N,
                                    k.threshold,type="Laplace",
                                    signe_tau = signe_taus[i_t])
}
```

Observe results

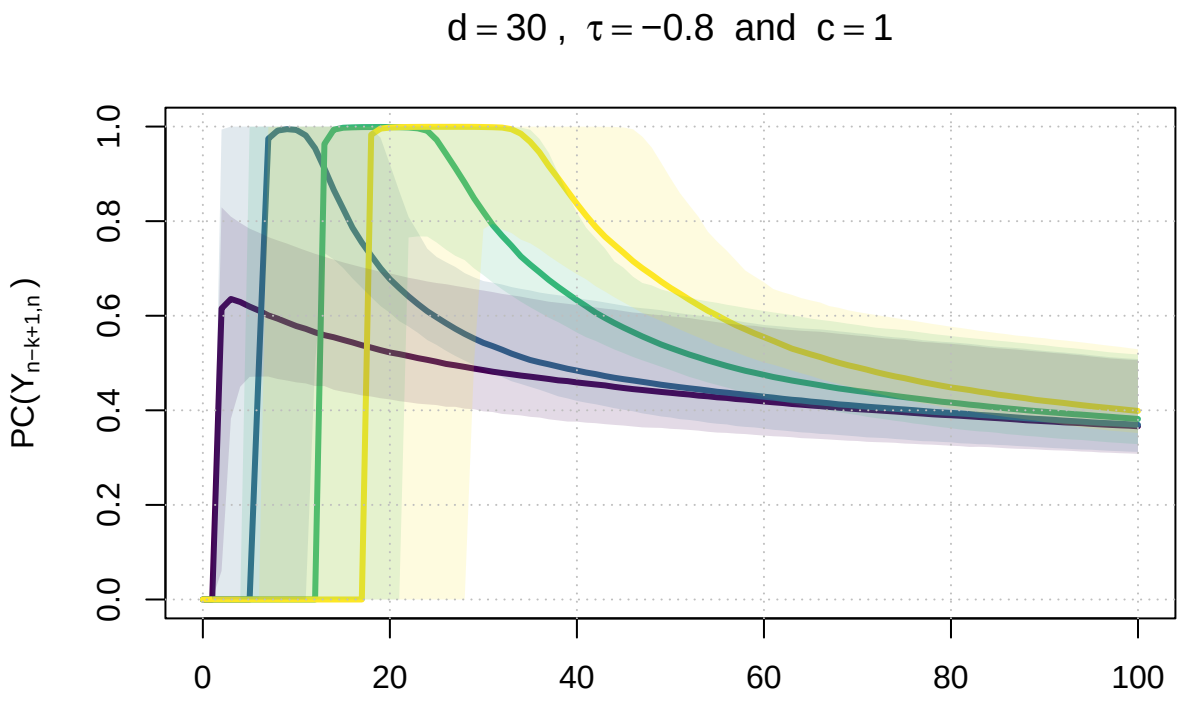
Use the following function

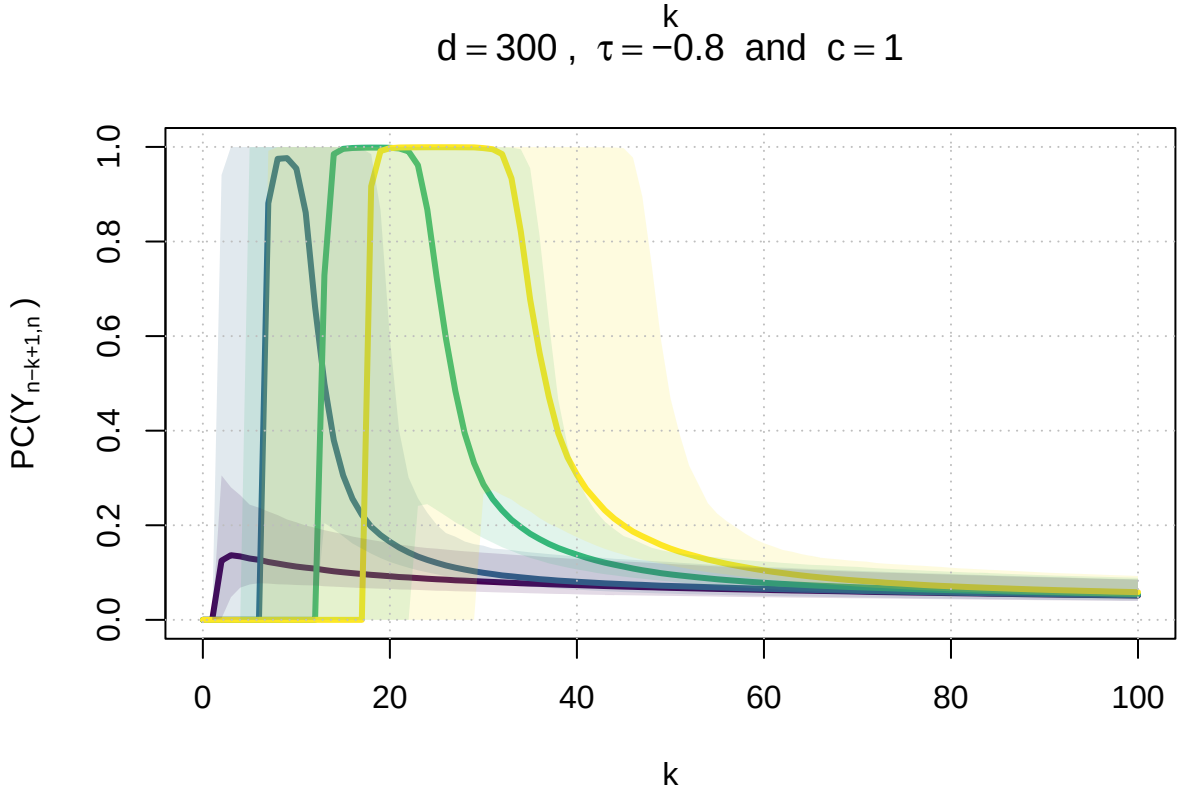
```
observe_sparse <- function(j){
  for(i_t in 1:length(thetas)){
    cos.rep_1 <- data.frame(results_sparse_p_1[[i_t]])
    cos.rep_2 <- data.frame(results_sparse_p_2[[i_t]])
    oo_beta_1=oo_beta_2 <- NULL
    file_1 <- paste("../SEPaLS_simulations/results/Laplace/simu_Laplace_c",j,
                    "_Theta",i_t,
                    "_mu1.pdf",sep="")
    file_2 <- paste("../SEPaLS_simulations/results/Laplace/simu_Laplace_c",j,
                    "_Theta",i_t,
                    "_mu2.pdf",sep="")
    for(i in 1:n_lambdas){
      id <- which(cos.rep_1$lambda==i &
                    cos.rep_1$id_power==j)
      if(i==1){
        oo_beta_1_median <- cos.rep_1$median[id]
        oo_beta_2_median <- cos.rep_2$median[id]
        oo_beta_1_quant5 <- cos.rep_1$quant5[id]
        oo_beta_2_quant5 <- cos.rep_2$quant5[id]
        oo_beta_1_quant95 <- cos.rep_1$quant95[id]
        oo_beta_2_quant95 <- cos.rep_2$quant95[id]
      }
      else
```

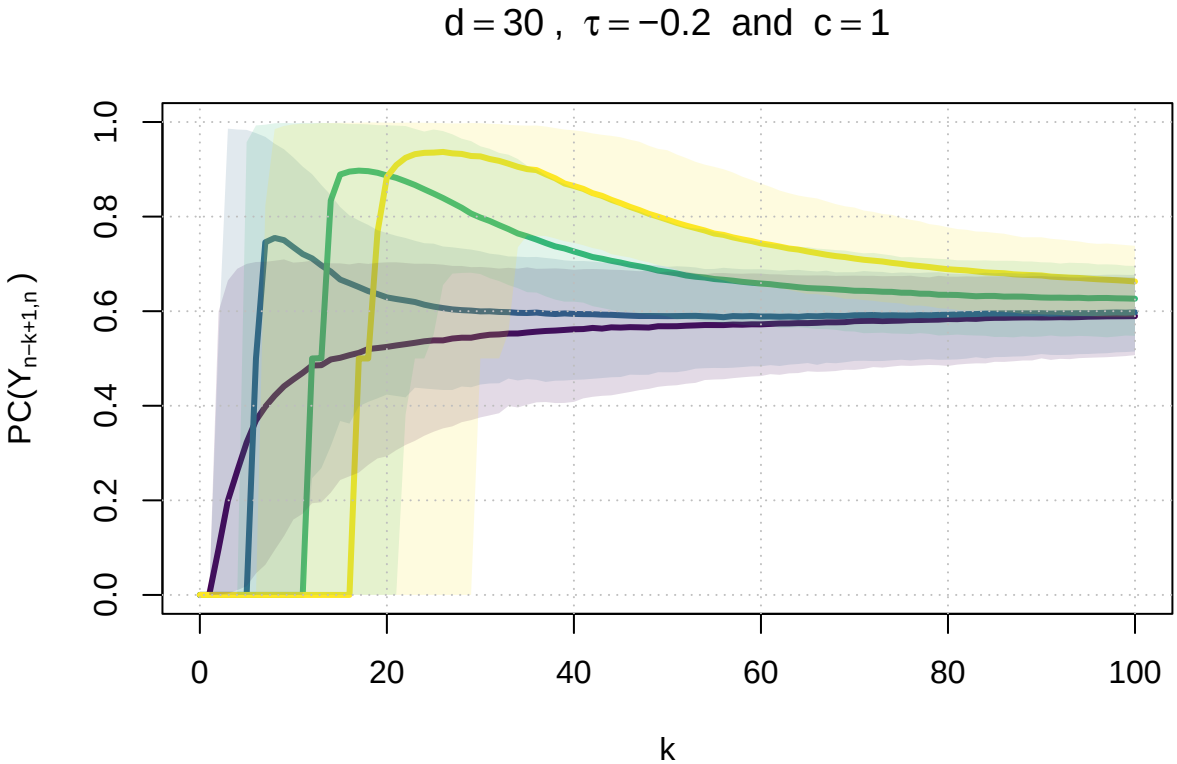
```
    {
      oo_beta_1_median <- cbind(oo_beta_1_median,
                                cos.rep_1$median[id])
      oo_beta_2_median <- cbind(oo_beta_2_median,
                                cos.rep_2$median[id])
      oo_beta_1_quant5 <- cbind(oo_beta_1_quant5,
                                cos.rep_1$quant5[id])
      oo_beta_2_quant5 <- cbind(oo_beta_2_quant5,
                                cos.rep_2$quant5[id])
      oo_beta_1_quant95 <- cbind(oo_beta_1_quant95,
                                 cos.rep_1$quant95[id])
      oo_beta_2_quant95 <- cbind(oo_beta_2_quant95,
                                 cos.rep_2$quant95[id])
    }
  }
  # pdf(file_1,width = 7,height = 5,onefile = T)
  XX <- n-unique(cos.rep_1$nb_exceed)
  main <- bquote(d==30~", "~tau==.(taus[i_t])~" and "~c==.(c_s[j]))
  matplot(XX,oo_beta_1_median,
          type="l",lty=1,lwd=3,main=main,
          ylab=bquote("PC(Y"["n-k+1,n"]~")"),
          xlab=expression(k),
          ylim=c(0,1),
          col=colors)
  for(jj in 1:ncol(oo_beta_1_quant5)){
    polygon(x=c(XX,rev(XX)),
            y=c(oo_beta_1_quant5[,jj],rev(oo_beta_1_quant95[,jj])),
            col=scales::alpha(colors[jj],alpha_transpa),border=NA)
  }
  abline(h=(0:5)/5,col="gray",lty=3)
  abline(v=(0:10)*20,col="gray",lty=3)
  # dev.off()
  # pdf(file_2,width = 7,height = 5,onefile = T)
  main <- bquote(d==300~", "~tau==.(taus[i_t])~" and "~c==.(c_s[j]))
  matplot(XX,oo_beta_2_median,
          type="l",lty=1,lwd=3,main=main,
          ylab=bquote("PC(Y"["n-k+1,n"]~")"),
          xlab=expression(k),
          ylim=c(0,1),
          col=colors)
  for(jj in 1:ncol(oo_beta_1_quant5)){
    polygon(x=c(XX,rev(XX)),
            y=c(oo_beta_2_quant5[,jj],rev(oo_beta_2_quant95[,jj])),
            col=scales::alpha(colors[jj],alpha_transpa),border=NA)
  }
  abline(h=(0:5)/5,col="gray",lty=3)
  abline(v=(0:10)*20,col="gray",lty=3)
  # dev.off()
 }
}
```

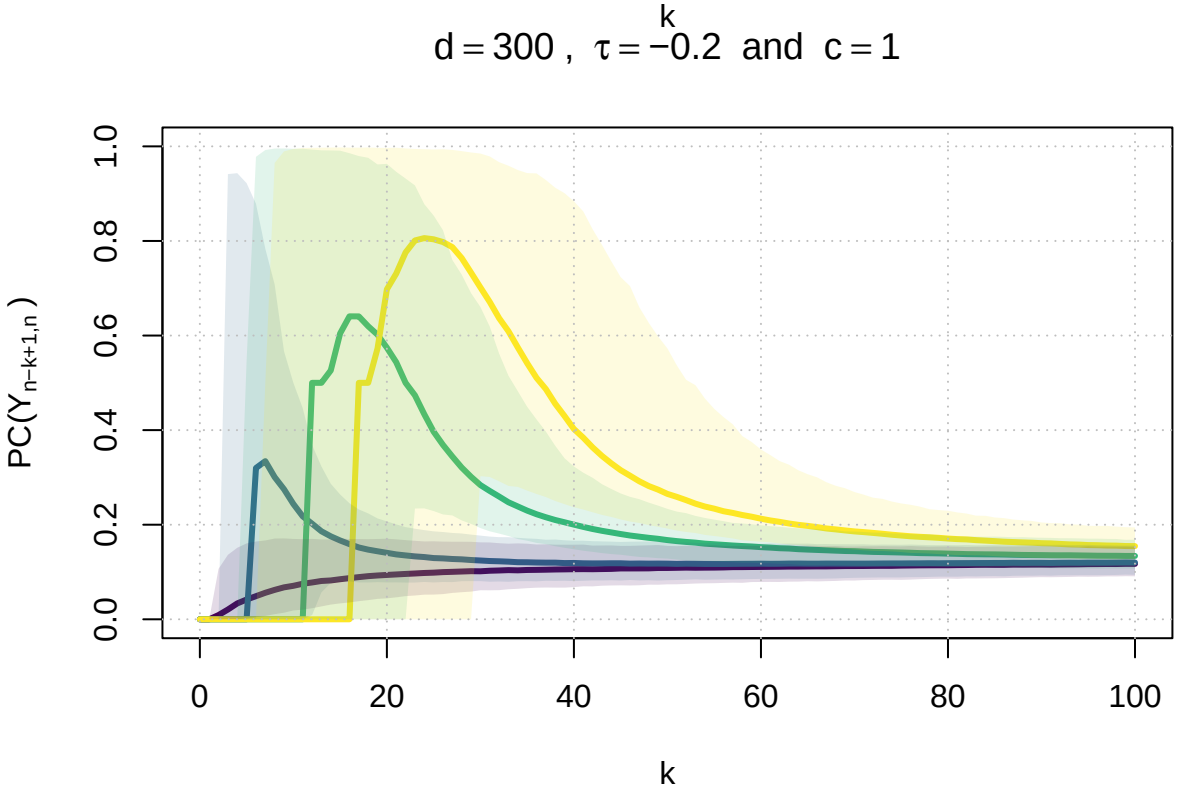
For $c = 1$

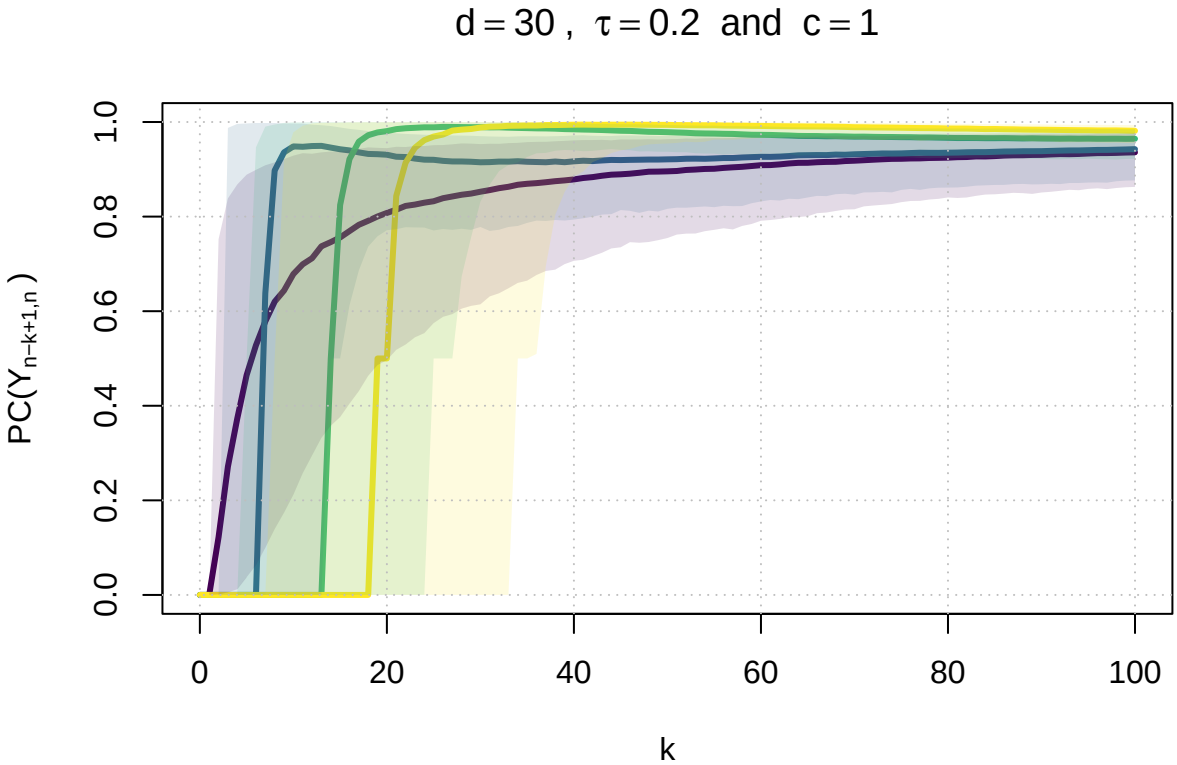
```
observe_sparse(which(c_s==1))
```

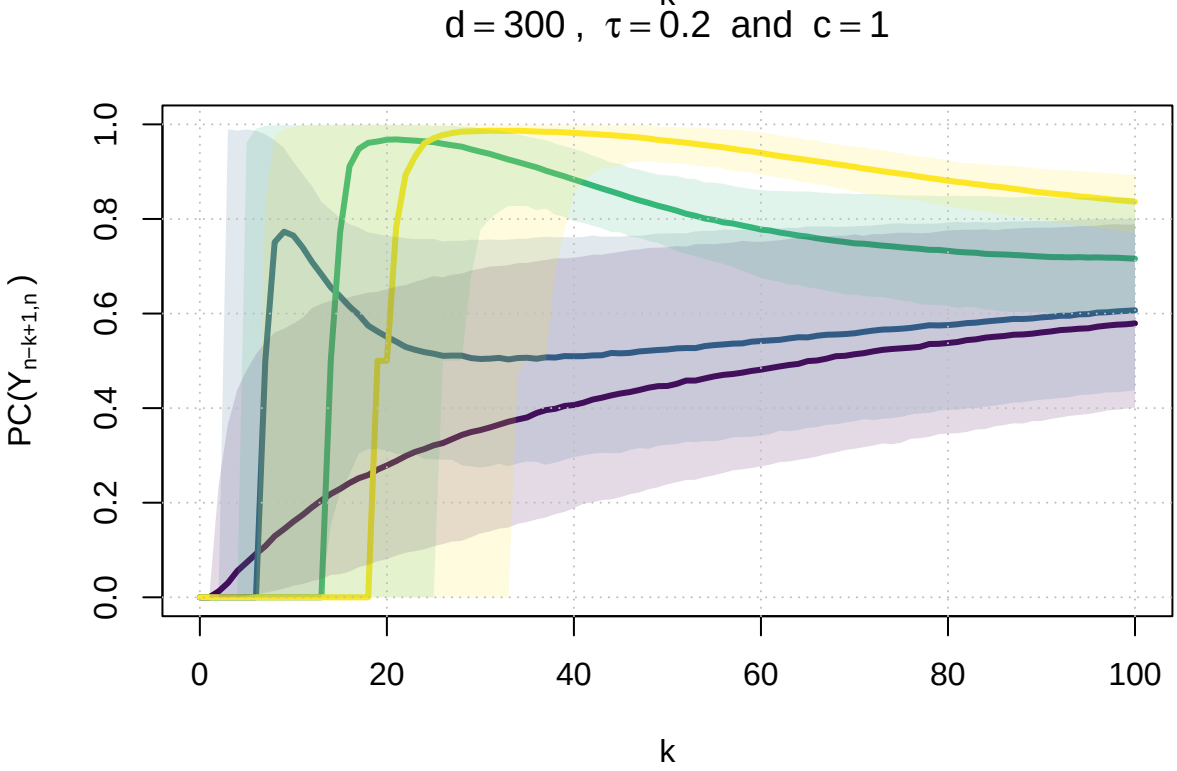


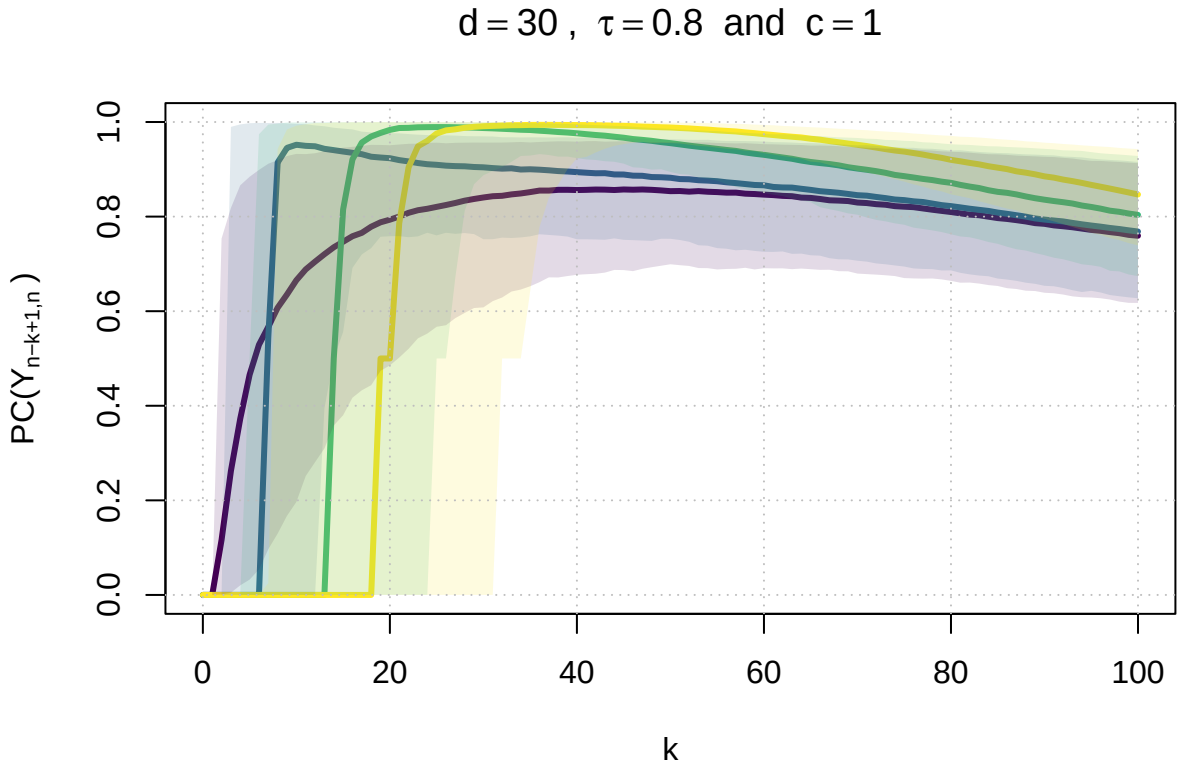


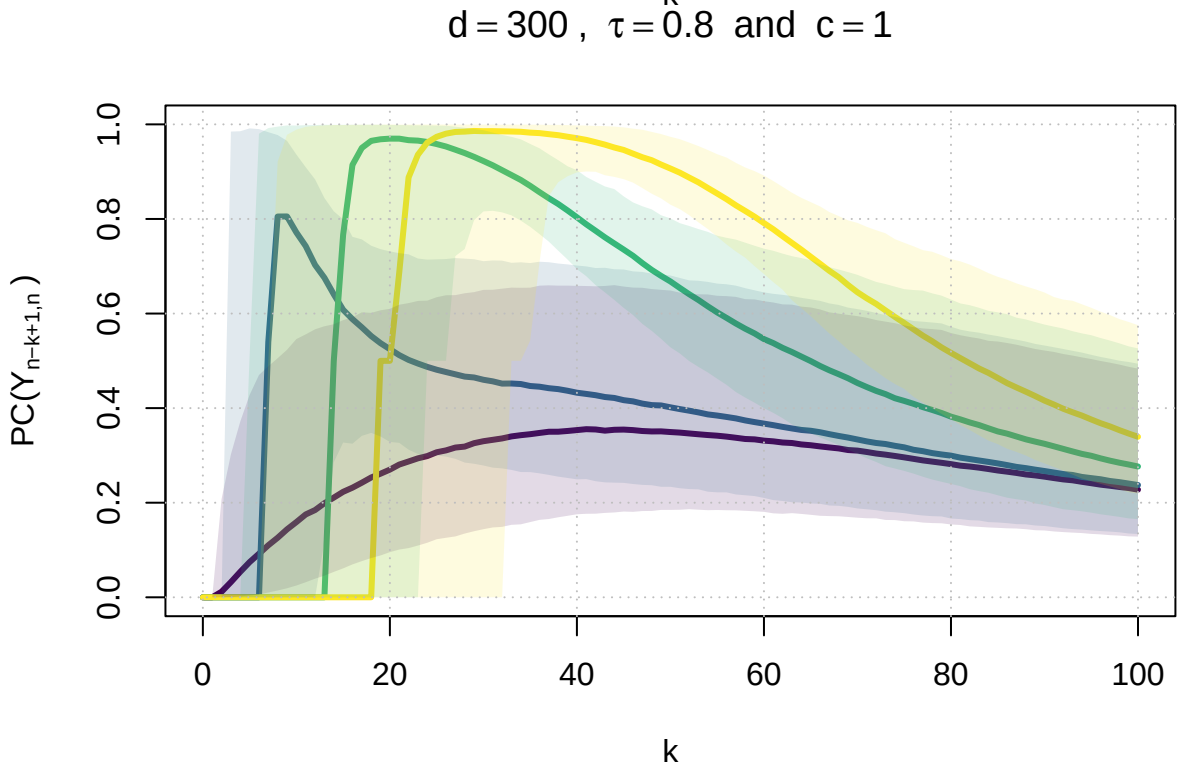

d = 30 , τ = −0.2 and c = 1
PC(Y n−k+1,n)
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
k


d = 300 , τ = −0.2 and c = 1
PC(Y n−k+1,n)
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
k


d = 30 , τ = 0.2 and c = 1
PC(Y n−k+1,n)
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
k

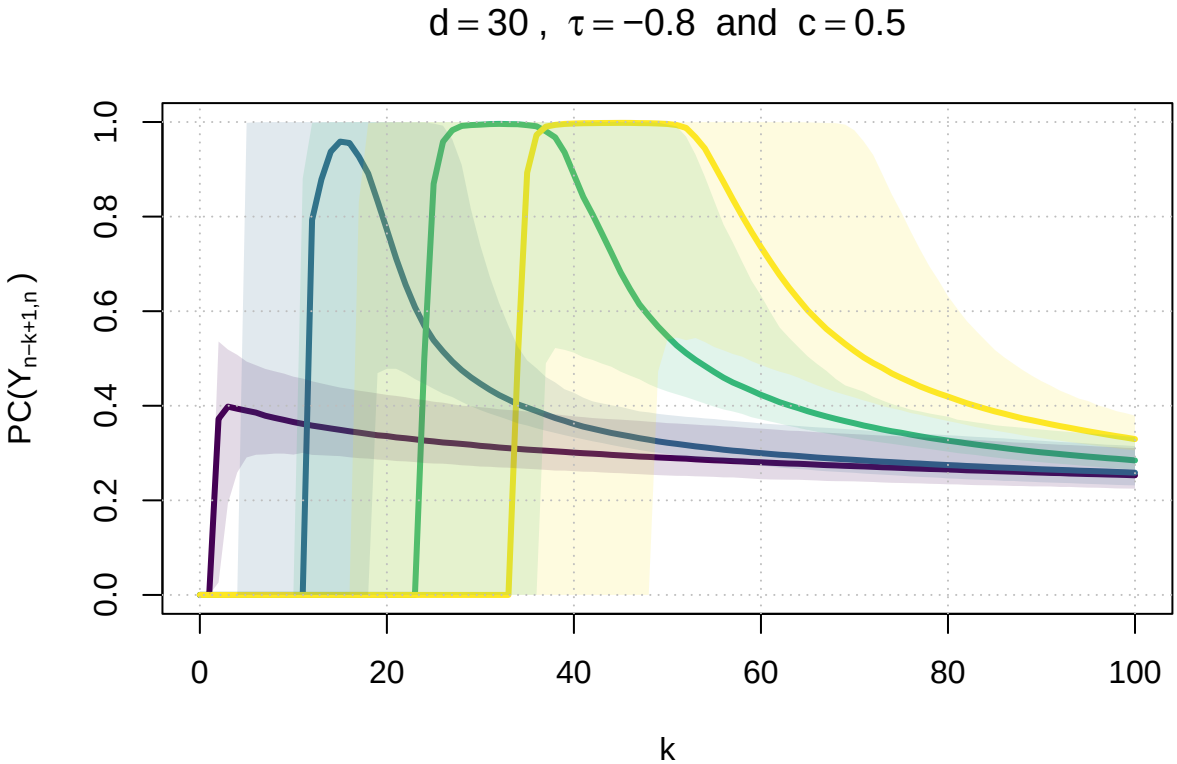

d = 300 , τ = 0.2 and c = 1
PC(Y n−k+1,n)
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
k

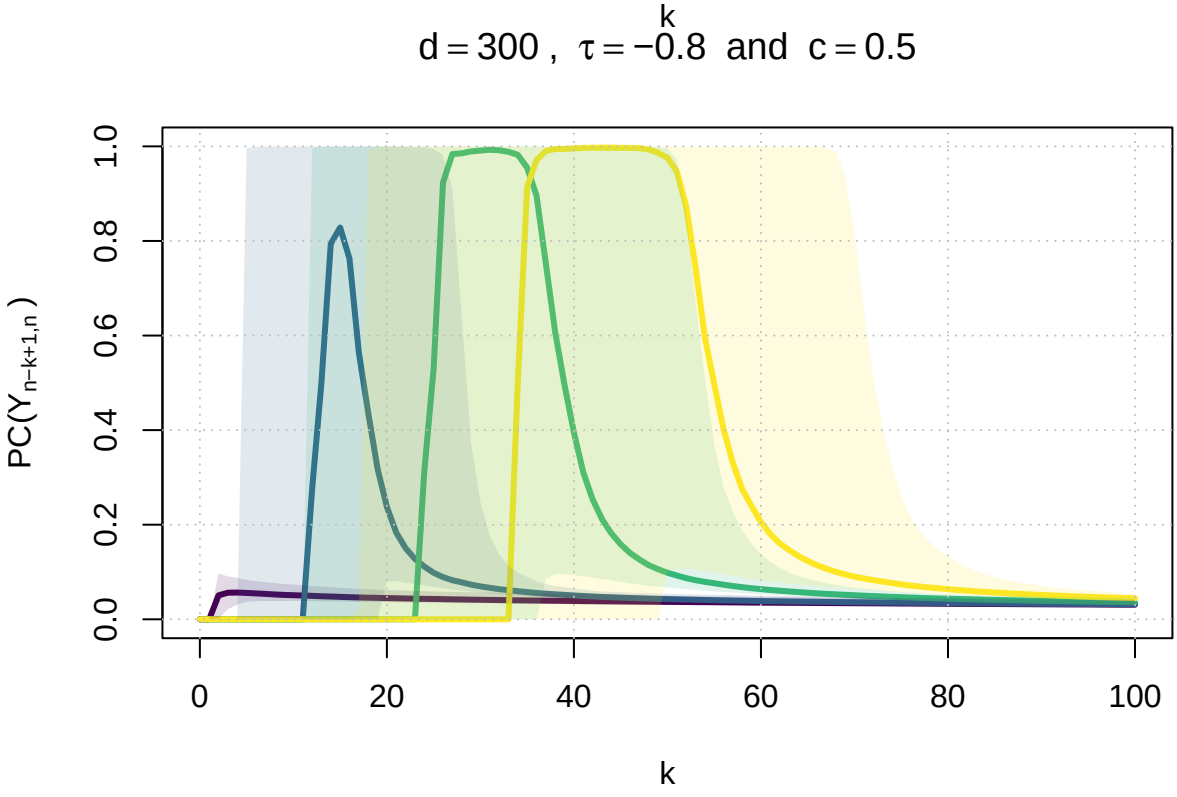


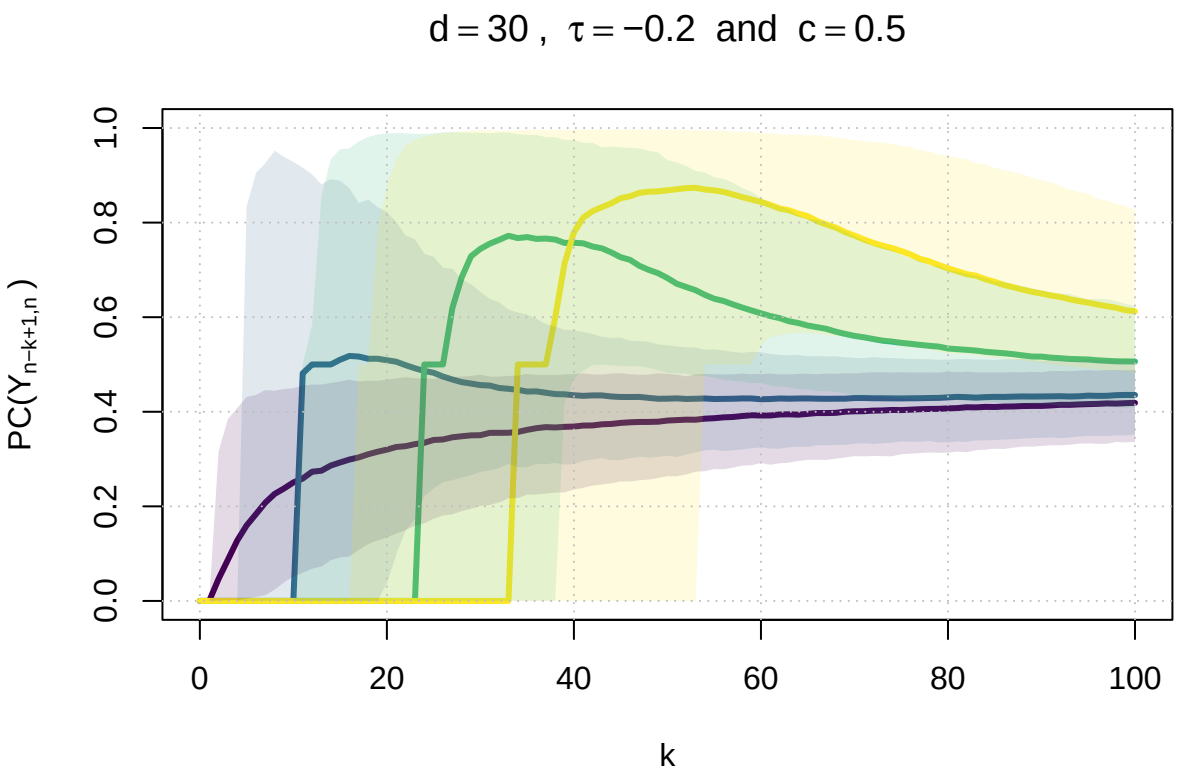


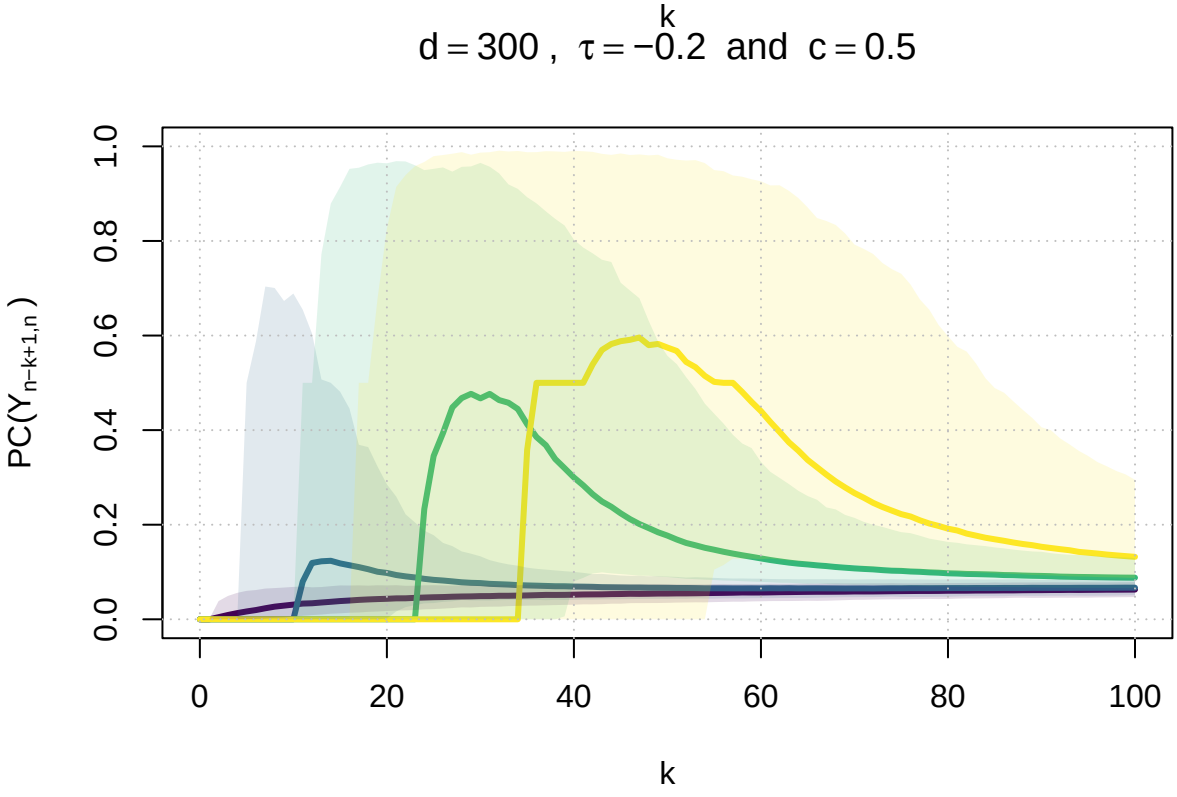
For $c = 1/2$

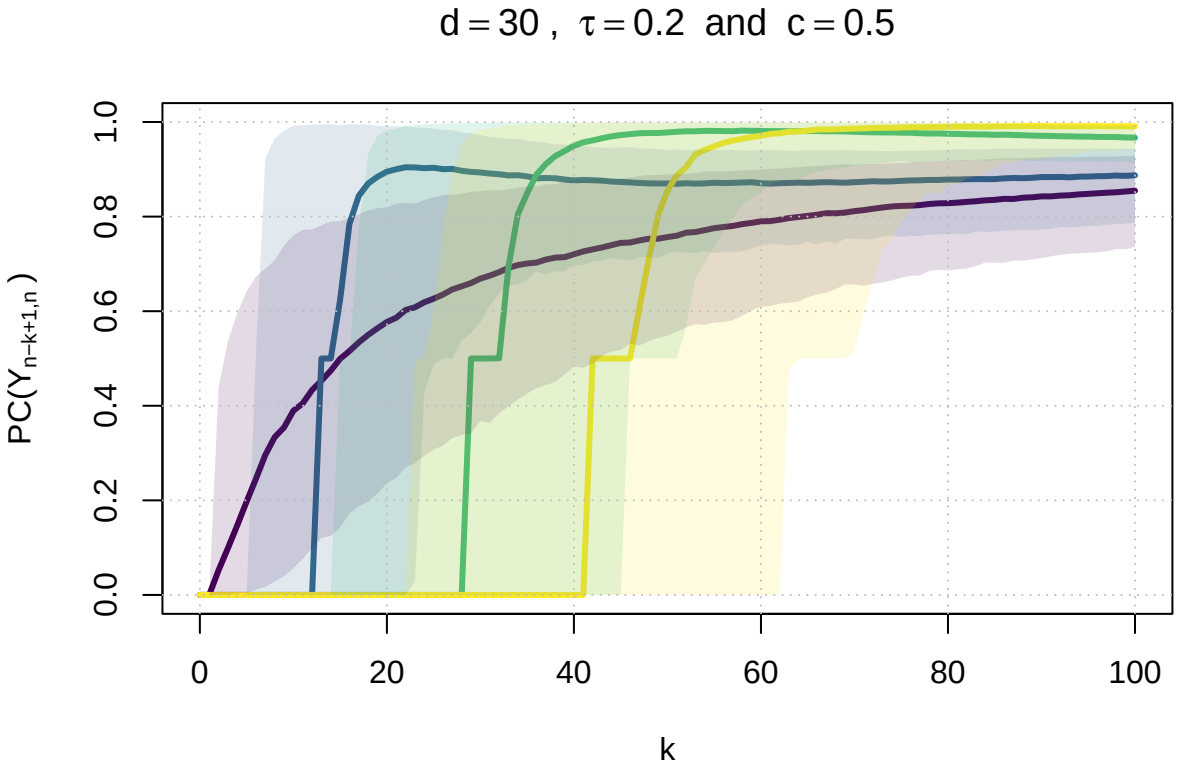
```
observe_sparse(which(c_s==1/2))
```

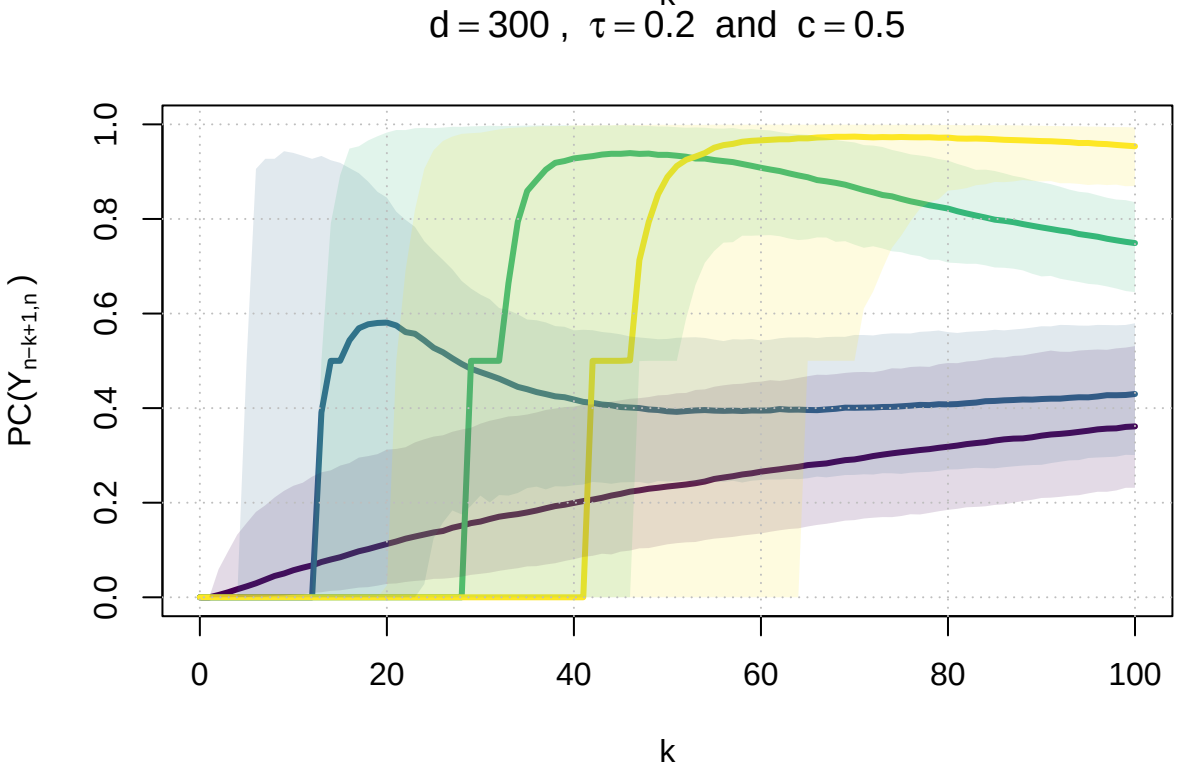

d = 30 , τ = −0.8 and c = 0.5
PC(Y n−k+1,n)
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
k

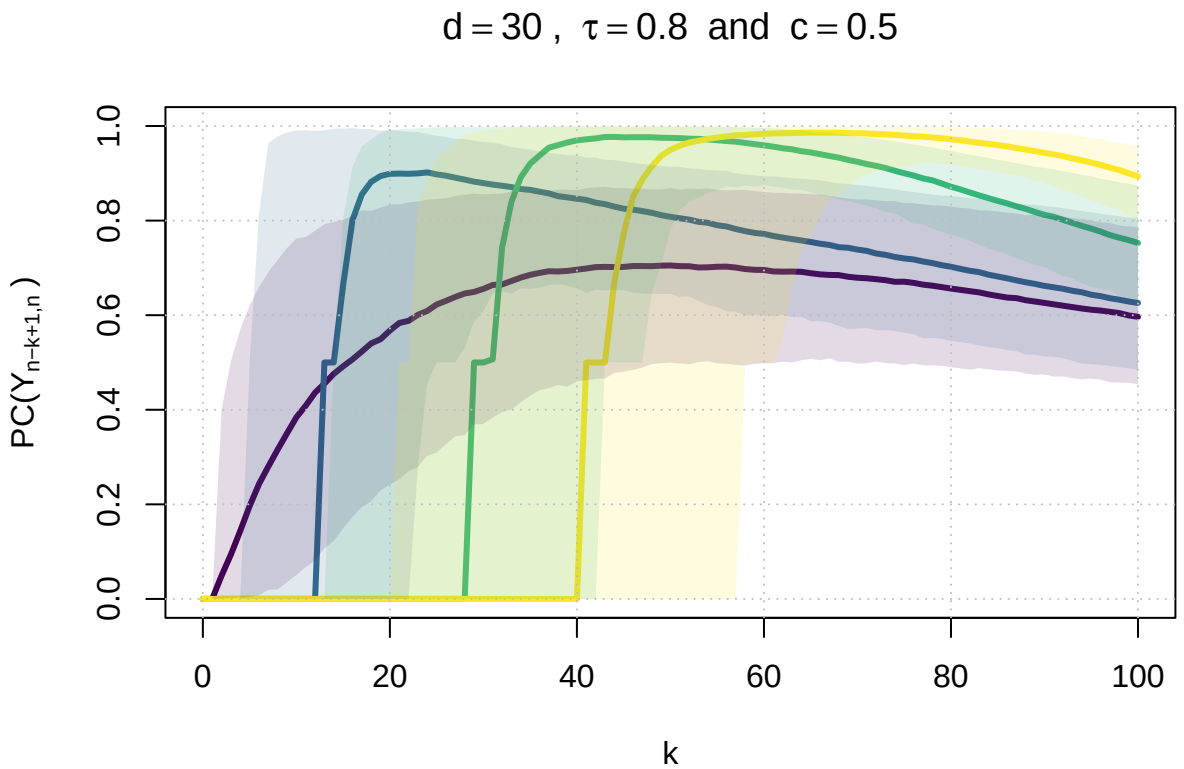

d = 300 , τ = −0.8 and c = 0.5
PC(Y n−k+1,n)
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
k

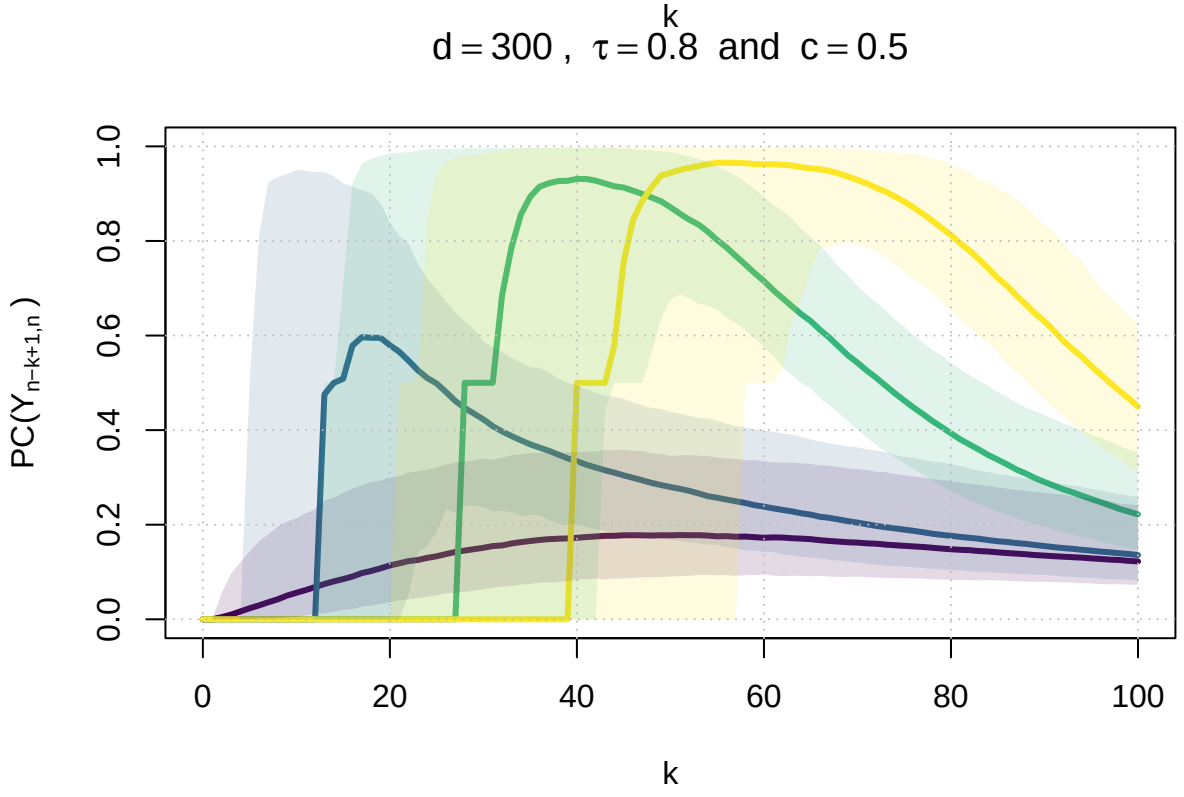

d = 30 , τ = −0.2 and c = 0.5
PC(Y n−k+1,n)
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
k


d = 300 , τ = −0.2 and c = 0.5
PC(Y n−k+1,n)
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
k


d = 30 , τ = 0.2 and c = 0.5
PC(Y n−k+1,n)
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
k

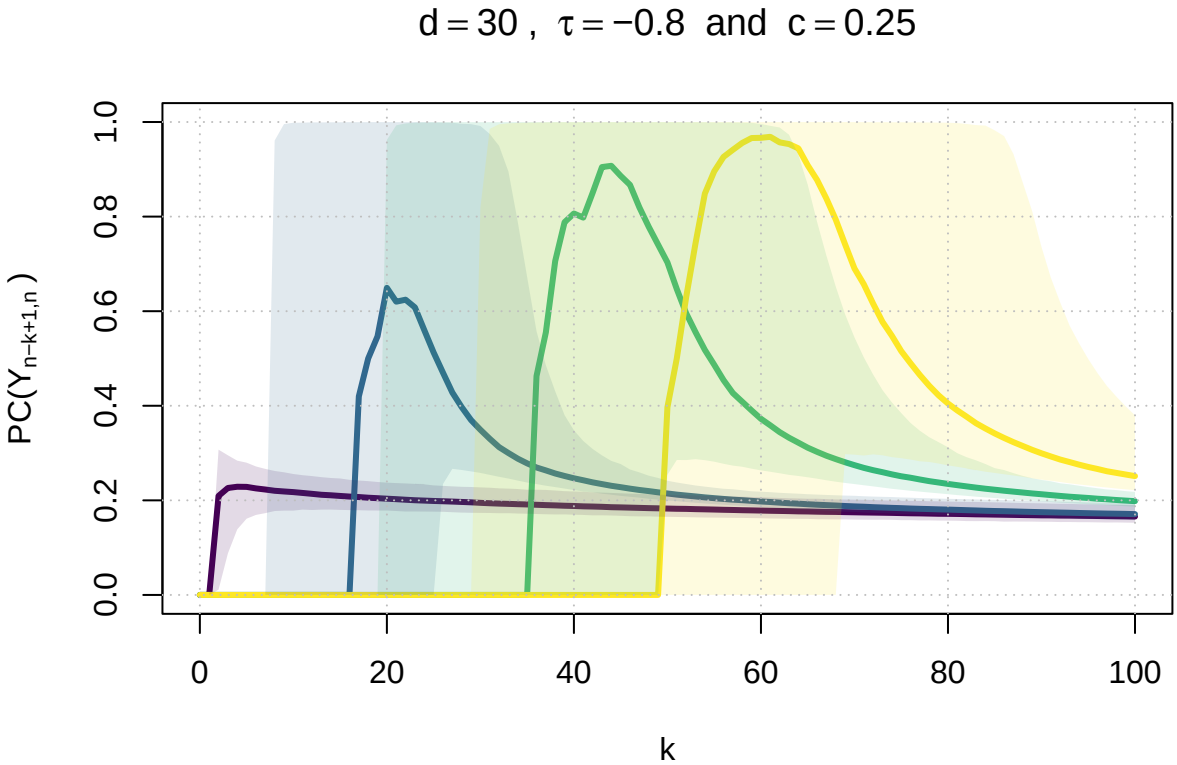

d = 300 , τ = 0.2 and c = 0.5
PC(Y n−k+1,n)
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
k

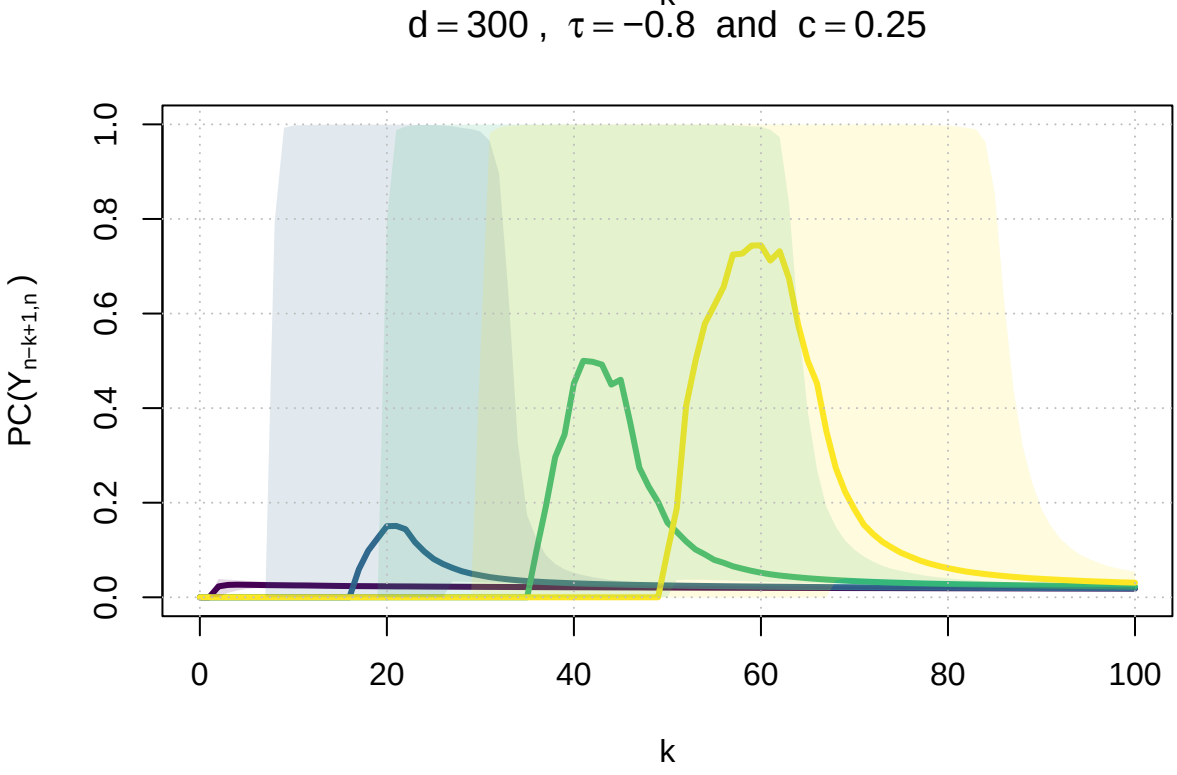


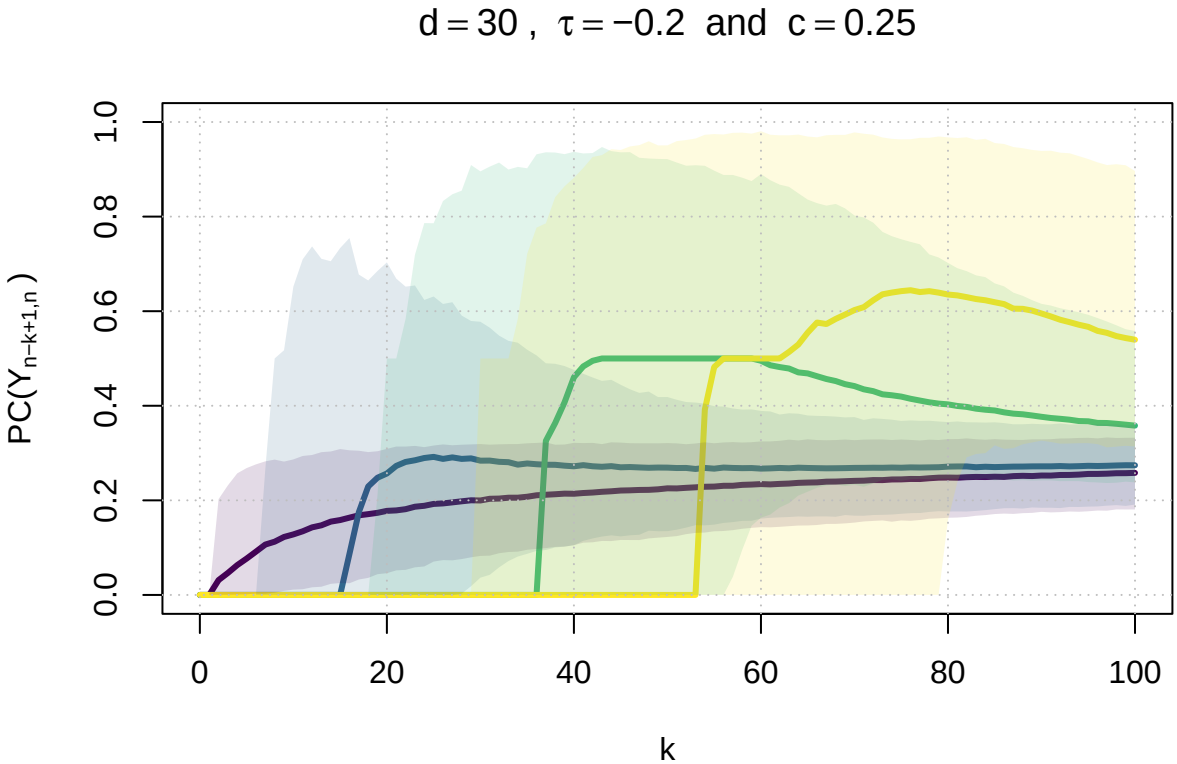


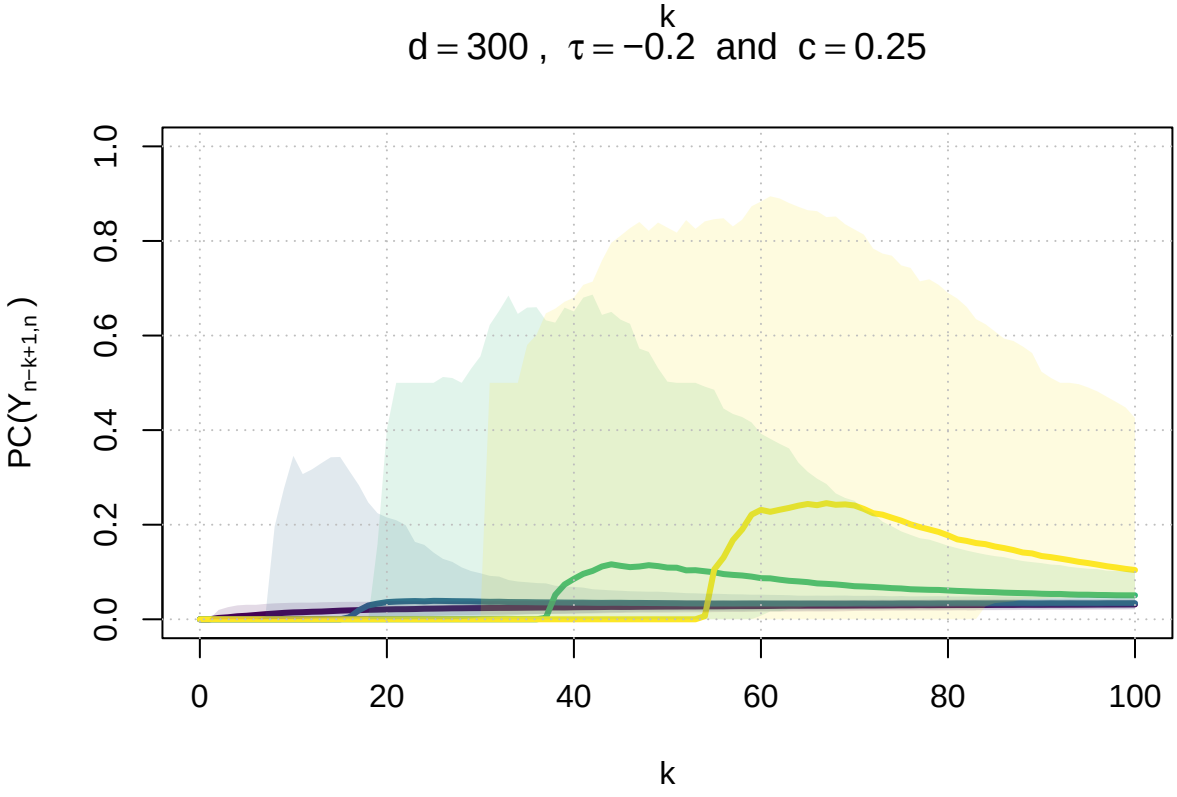
For $c = 1/4$

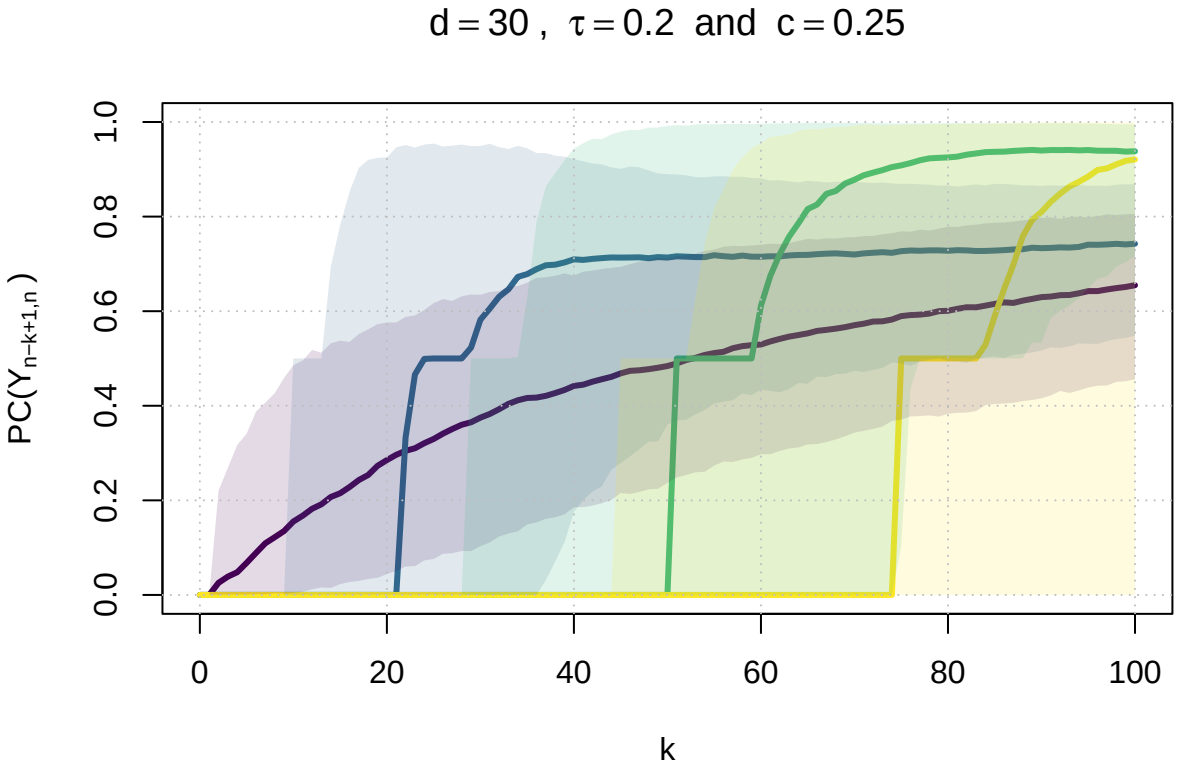
```
observe_sparse(which(c_s==1/4))
```

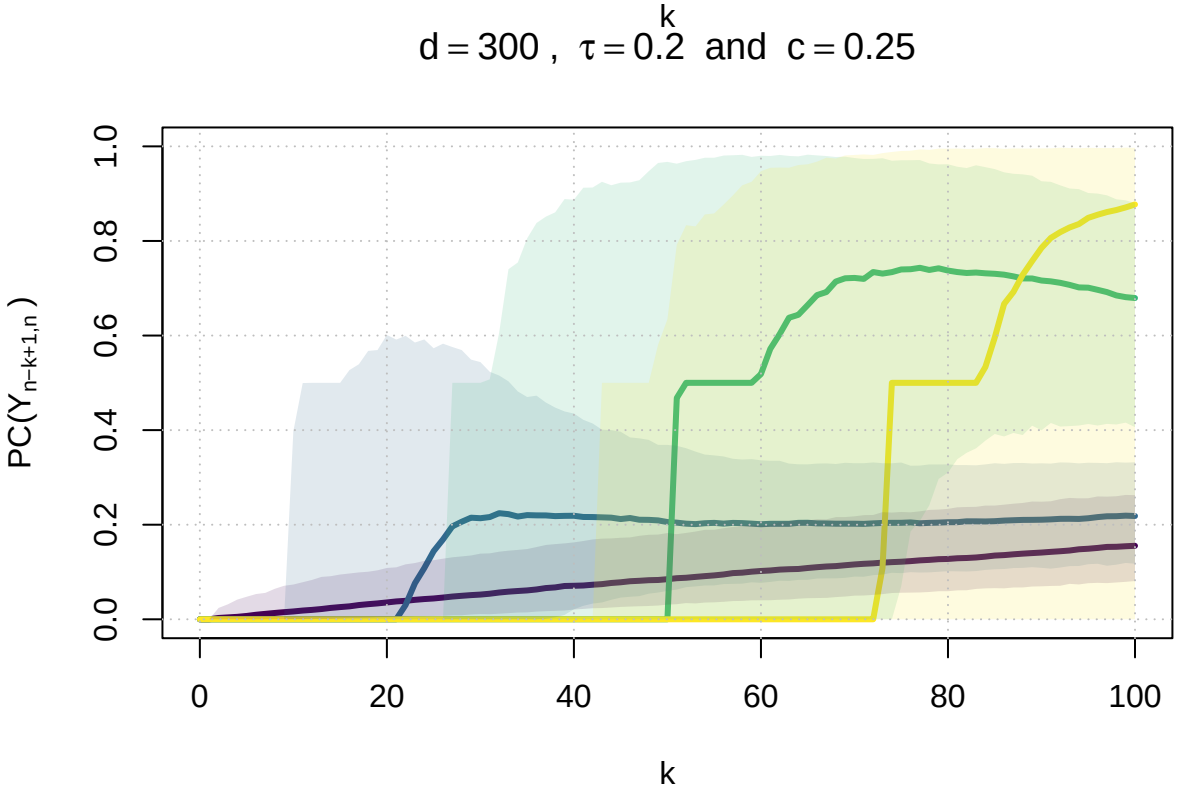

d = 30 , τ = −0.8 and c = 0.25
PC(Y n−k+1,n)
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
k

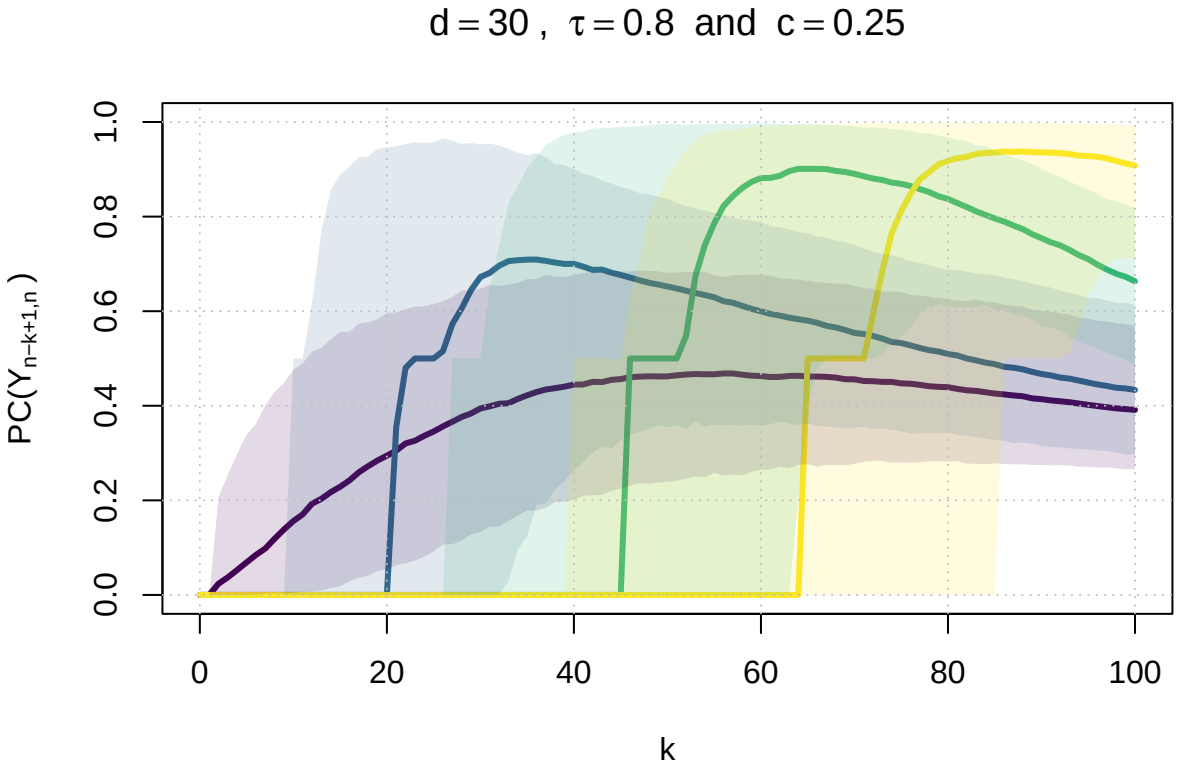

d = 300 , τ = −0.8 and c = 0.25
PC(Y n−k+1,n)
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
k

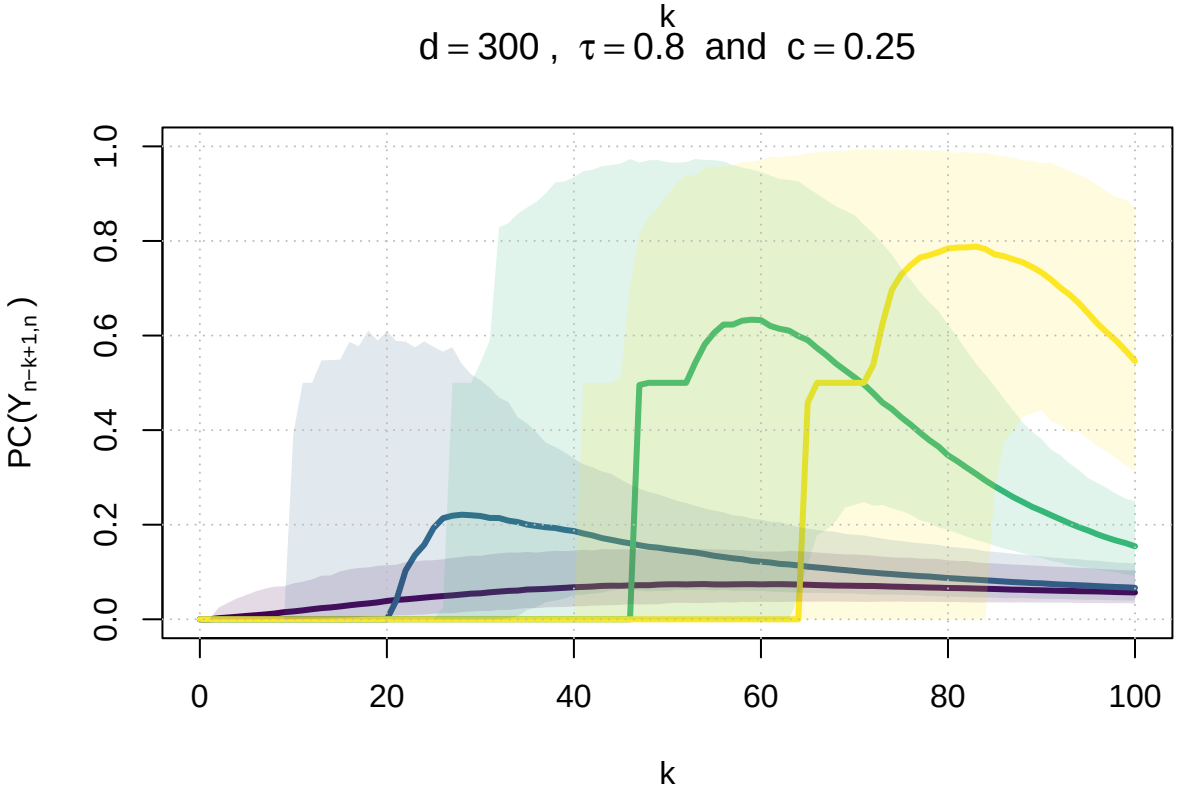

d = 30 , τ = −0.2 and c = 0.25
PC(Y n−k+1,n)
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
k


d = 300 , τ = −0.2 and c = 0.25
PC(Y n−k+1,n)
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
k


d = 30 , τ = 0.2 and c = 0.25
PC(Y n−k+1,n)
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
k


d = 300 , τ = 0.2 and c = 0.25
PC(Y n−k+1,n)
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
k


d = 30 , τ = 0.8 and c = 0.25
PC(Y n−k+1,n)
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
k


d = 300 , τ = 0.8 and c = 0.25
PC(Y n−k+1,n)
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
k