NATIONAL RESEARCH UNIVERSITY

HIGHER SCHOOL OF ECONOMICS

Faculty of Economic Sciences

“Statistical modelling and actuarial science”

HOME ASSIGNMENT #1

Bogomolova Polina

Group MSM171

Moscow

2018

Numerical part

##N1##

```
presidents=presidents
```

#i#

```
Hist_Sturges<-hist.default(presidents, breaks="Sturges",
col="bisque1")

length(Hist_Sturges$counts)

ceiling(log2(120))+1

pretty(presidents, 8)

library(zoo)

presidents_new<-na.approx(presidents)

presidents_new

Hist_Sturges<-hist.default(presidents_new, breaks="Sturges",
col="bisque1")

length(Hist_Sturges$counts)
```

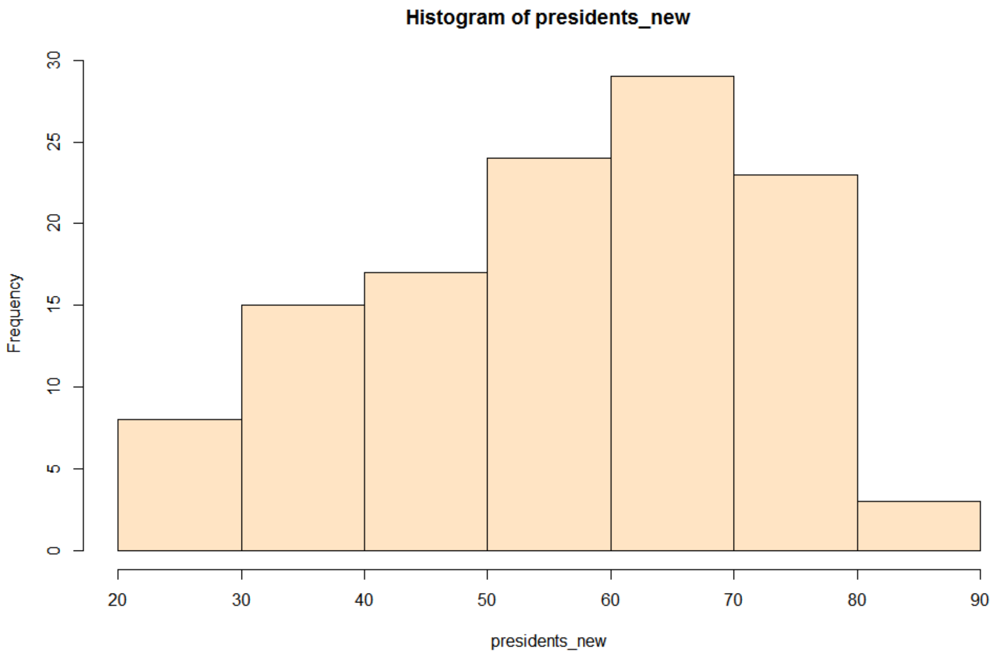

Figure 1. Histogram with amount of bins selected by Sturges rule

#ii#

```
kde_g<-density(presidents_new, kernel="gaussian")

kde_e<-density(presidents_new, kernel="epanechnikov")

kde_r<-density(presidents_new, kernel="rectangular")
```

```
kde_t<-density(presidents_new, kernel="triangular")
kde_b<-density(presidents_new, kernel="biweight")
kde_c<-density(presidents_new, kernel="cosine")
kde_oc<-density(presidents_new, kernel="optcosine")
plot(kde_g, col="red",xlim=c(-30,110),ylim=c(0,0.05), main="Kernels")
lines(kde_e, col="blue")
lines(kde_r, col="green")
lines(kde_t, col="yellow")
lines(kde_b, col="brown")
lines(kde_c, col="pink")
lines(kde_oc, col="orange")
legend("topleft",legend=c("gaussian","epanechnikov","rectangular",
       "triangular","biweighted","cosine","optcosine"),
       col=c("red","blue","green","yellow","brown","pink","orange"),
       lty=1:7, title="Kernel estimates")
```

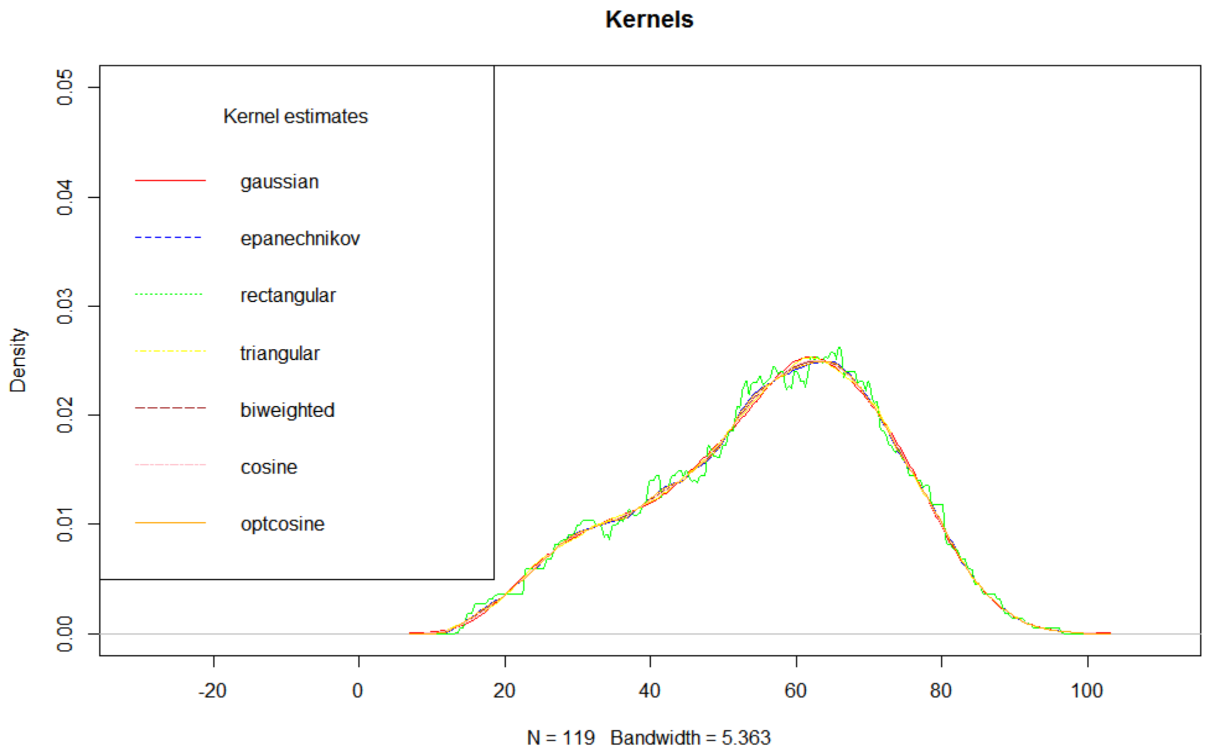

Figure 2. Kernel estimators with various kernels

```
#iii#
bw.nrd0(presidents_new)
bw.nrd(presidents_new)
```

```
bw.ucv(presidents_new)
bw.bcv(presidents_new)
bw.SJ(presidents_new, method="ste")
bw.SJ(presidents_new, method="dpi")
kde1<-density(presidents_new, bw="nrd0")
kde2<-density(presidents_new, bw="nrd")
kde3<-density(presidents_new, bw="ucv")
kde4<-density(presidents_new, bw="bcv")
kde5<-density(presidents_new, bw="SJ-ste")
kde6<-density(presidents_new, bw="SJ-dpi")
plot(kde1, col="red",xlim=c(-40,110),ylim=c(0,0.05), main="Kernels")
lines(kde2, col="blue")
lines(kde3, col="green")
lines(kde4, col="yellow")
lines(kde5, col="brown")
lines(kde6, col="pink")
legend("topleft",legend=c("Silverman","Scott","unbiased CV",
     "biased CV","Sheather&Jones-ste","Sheather&Jones-dpi"),
     col=c("red","blue","green","yellow","brown","pink"),
     lty=1:6, title="Bandwidth selection method")
```

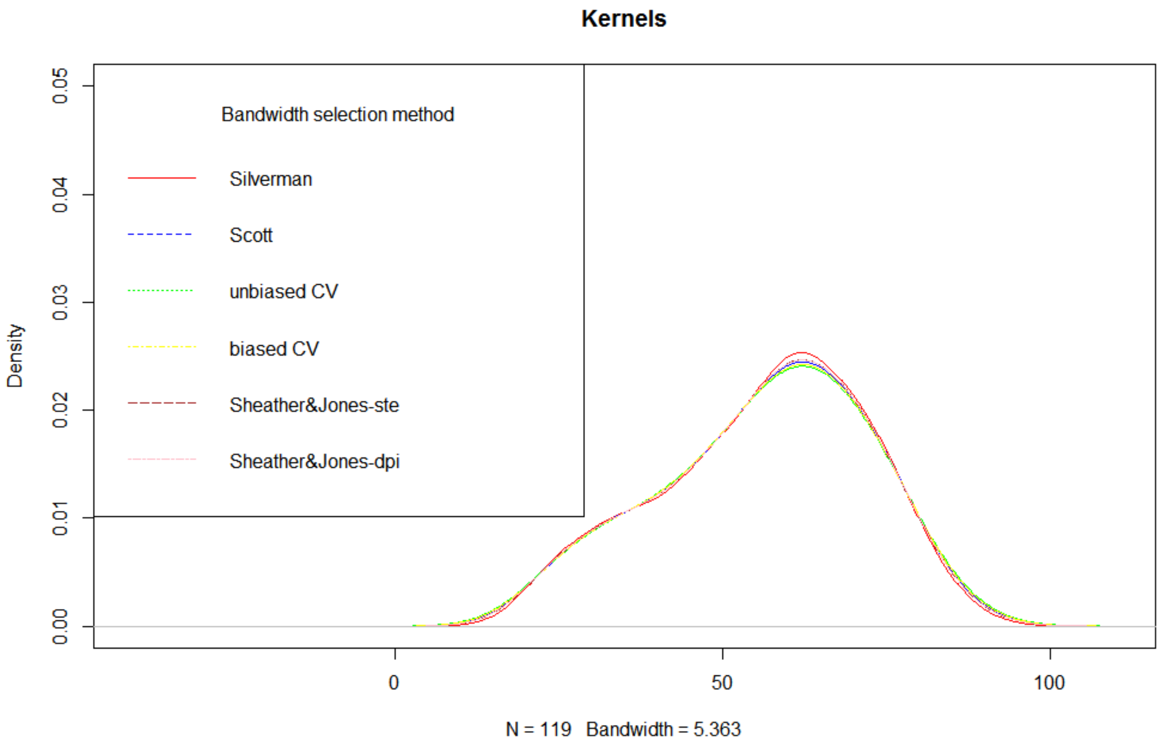

Figure 3. Kernel estimators under various choices of bandwidth

```
#iv#
hist(presidents_new, freq=FALSE, col="bisque1",
     main="Histogram and kernels", xlim=c(-50,100))
lines(kde2, col="blue")
lines(kde3, col="green")
lines(kde4, col="yellow")
lines(kde5, col="brown")
lines(kde6, col="pink")
legend("topleft",legend=c("Silverman","Scott","unbiased CV",
        "biased CV","Sheather&Jones-ste","Sheather&Jones-dpi"),
       col=c("red","blue","green","yellow","brown","pink"),
       lty=1:6, title="Bandwidth selection method")
density1<-density(presidents_new)
kerneltype=c("gaussian", "epanechnikov", "rectangular",
          "triangular", "biweight", "cosine", "optcosine")
bw=c("nrd", "nrd0", "ucv", "bcv", "SJ-ste", "SJ-dpi")
x=function(density2){
  x=0
  for (j in (1:length(density2$y))){
    x=x+(density2$y[j]-density1$y[j])^2
  }
  return(x)
}
e=c(rep(0,42))
l=1
for (i in (1:7)){
  for (j in (1:6)){
    e[l]=(x(density(presidents_new, kernel=kerneltype[i],
                    bw=bw[j])))
    l=l+1
  }
}
index<-which(e==min(e))
bw[index%/%6+1]
```

```
kerneltype[index%%6]
```

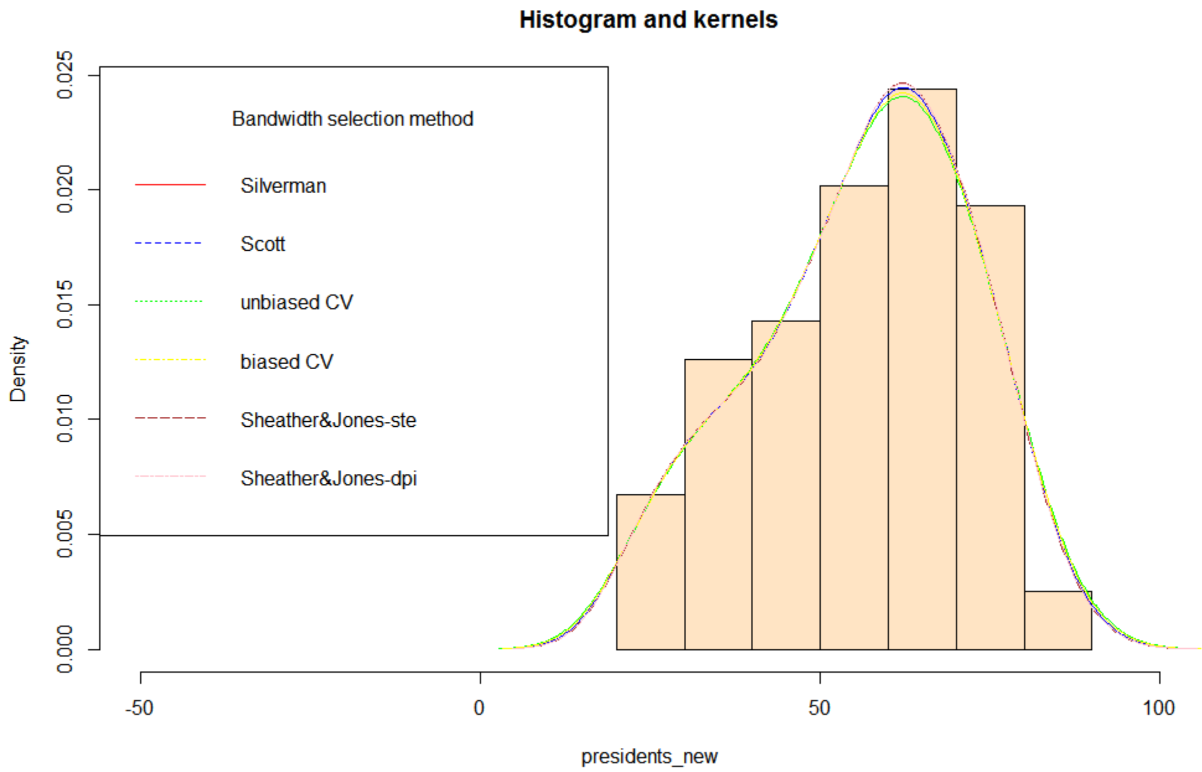

Figure 4. Histogram with kernel estimators

Result:

```
> bw[index%/%6+1]
[1] "nrd"
> kerneltype[index%%6]
[1] "epanechnikov"
```

```
##N2##
#i#
N=1000
eta=sample(1:3, N, TRUE, prob = c(1/2,1/4,1/4))
hist(eta, breaks=c(0.5, 1.5, 2.5, 3.5),probability=TRUE,
col="lightblue3")
f_n = 1
x_make=function(){
for (i in (1:N)){
  ifelse(eta[i]=="1",f_n[i]<-rnorm(1, mean=0, sd=1),
```

```
        ifelse(eta[i]=="2", f_n[i]<-(rexp(1, rate=1)-1),
               f_n[i]<-(-rexp(1, rate=1)+1)))}
  return(f_n)
}
f_n=x_make()
hst<-hist(f_n, breaks="Sturges", col="lightblue3", probability=TRUE)
```

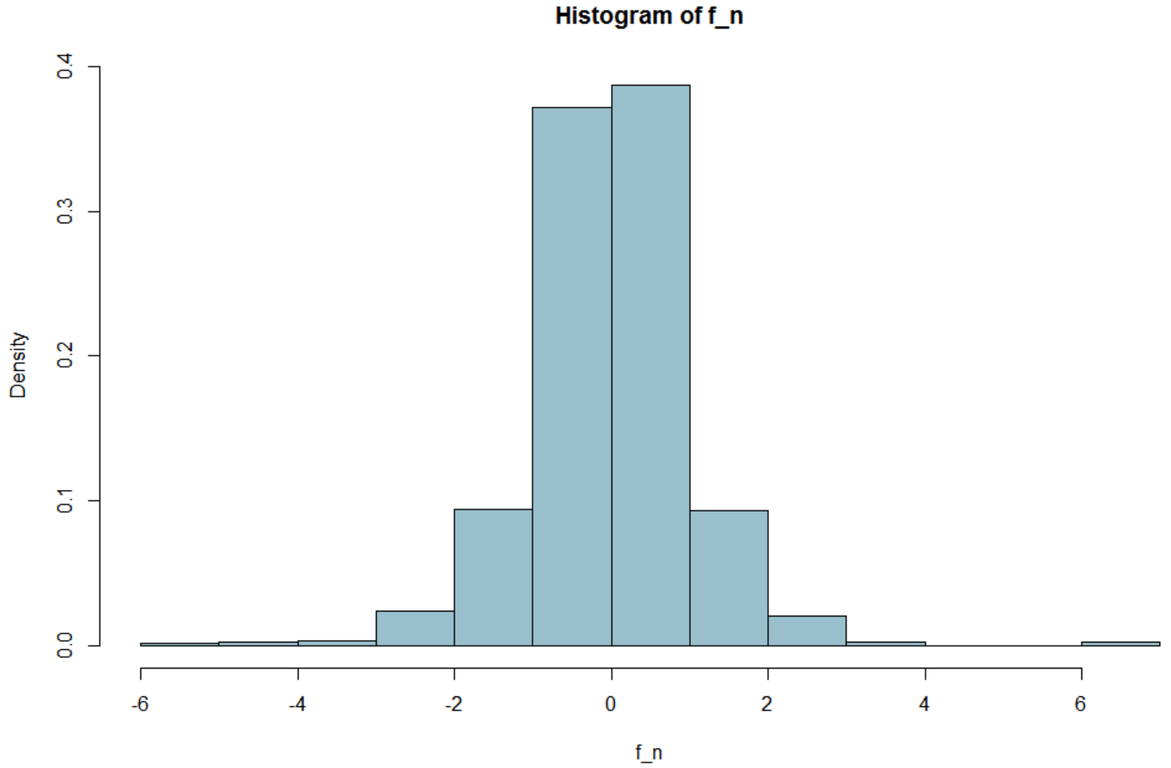

Figure 5. Histogram of the density p(x)

```
#ii#
Q=10000
x=seq(from=-3, to=3, length.out = Q)
whichdens<-
function(number){hst$density[max(which(hst$breaks<number))]}
p_x=function(x){
  p_x=1/2*dnorm(x, 0, 1)+1/4*(dexp(x+1, rate=1))+1/4*(dexp(-x+1,
rate=1))
  return(p_x)
}
mise=0
for (j in (1:Q)){
  mise=mise+(whichdens(x[j])-p_x(x[j]))^2
```

```
}
mise_min=1/Q*mise
mise_min
```

Result:

```
> mise_min
[1] 0.0004765514
```

```
#iii#
J=20
miseJ_min<-c(1:J)
for (k in (1:J)){
  x1<-x_make()
  hst1=hist(x1, breaks="Sturges", plot=FALSE)
  miseJ=0
  for (j in (1:length(x1))){
    miseJ=miseJ+(hst1$density[max(which(hst1$breaks<x1[j]))]-
p_x(x1[j]))^2
  }
  miseJ_min[k]=1/Q*miseJ
}
miseJ_min
```

Result:

```
> miseJ_min
 [1] 3.862508e-05 2.955752e-05 2.644167e-05 3.117355e-05 2.793692e-05
 [6] 3.912337e-05 4.853975e-05 2.725190e-05 2.695339e-05 4.131488e-05
[11] 2.925608e-05 3.134115e-05 4.234449e-05 4.000083e-05 3.333736e-05
[16] 2.967306e-05 3.191034e-05 3.874572e-05 5.624660e-05 6.944471e-05
```

```
#iv#
hst_Scott<-hist(f_n, breaks="Scott", col="lightblue3",
probability=TRUE)
```

```
whichdens_Scott<-
function(number){hst_Scott$density[max(which(hst_Scott$breaks<number))
]}

mise_Scott=0

for (j in (1:Q)){

  mise_Scott=mise_Scott+(whichdens_Scott(x[j])-p_x(x[j]))^2

}

mise_Scottmin=1/Q*mise_Scott

mise_Scottmin
```

Result:

> mise_Scottmin

[1] 0.0002584102

```
hst_FD<-hist(f_n, breaks="Freedman-Diaconis", col="lightblue3",
probability=TRUE)

whichdens_FD<-
function(number){hst_FD$density[max(which(hst_FD$breaks<number))]}

mise_FD=0

for (j in (1:Q)){

  mise_FD=mise_FD+(whichdens_FD(x[j])-p_x(x[j]))^2

}

mise_FDmin=1/Q*mise_FD

mise_FDmin
```

Result:

> mise_FDmin

[1] 0.0005468563

```
mises_min<-c(1:J)

for (k in (1:J)){

  xs<-x_make()

  hsts=hist(xs, breaks="Scott", plot=FALSE)

  mises=0

  for (j in (1:length(xs))){
```

```
    mises=mises+(hsts$density[max(which(hsts$breaks<xs[j]))]-
p_x(xs[j]))^2
   }
  mises_min[k]=1/Q*mises
}
mises_min
```

Result:

```
> mises_min
 [1] 6.796150e-05 2.461594e-05 1.665309e-04 2.217775e-05 7.342260e-05
 [6] 8.209588e-05 4.754425e-05 8.689651e-05 1.158622e-04 9.227986e-05
[11] 7.374253e-05 3.714309e-05 2.772159e-05 9.412061e-05 1.645007e-04
[16] 3.022039e-05 1.490534e-04 7.930696e-05 3.449422e-05 4.140765e-05
```

```
miseFD_min<-c(1:J)
for (k in (1:J)){
  xFD<-x_make()
  hstFD=hist(xFD, breaks="Freedman-Diaconis", plot=FALSE)
  miseFD=0
  for (j in (1:length(xFD))){
    miseFD=miseFD+(hstFD$density[max(which(hstFD$breaks<xFD[j]))]-
p_x(xFD[j]))^2
   }
  miseFD_min[k]=1/Q*miseFD
}
miseFD_min
```

Result:

```
> miseFD_min
 [1] 1.253671e-04 8.655710e-05 1.269549e-04 1.901296e-04 1.748743e-04
 [6] 1.930640e-04 1.089641e-04 3.286141e-04 1.590511e-04 1.107098e-04
[11] 1.018004e-04 1.623091e-04 6.574721e-05 2.031553e-04 1.584866e-04
[16] 1.558173e-04 1.853522e-04 6.727730e-05 1.103430e-04 2.082620e-04
```

```
#v#
bw<-seq(from=0.01, to=1, by=0.01)
mise_Stmin<-c(1:length(bw))
x2<-x_make()
for (i in (1:length(bw))){
  kde_Sturges<-density(x2, kernel="epanechnikov", bw=bw[i])
  mise_St=0
   for(j in (1:length(kde_Sturges$x))){
   mise_St=mise_St+(kde_Sturges$y[j]-p_x(kde_Sturges$x[j]))^2
}
mise_Stmin[i]=mise_St/length(kde_Sturges$x)
}
plot(bw, mise_Stmin, type="l")
```

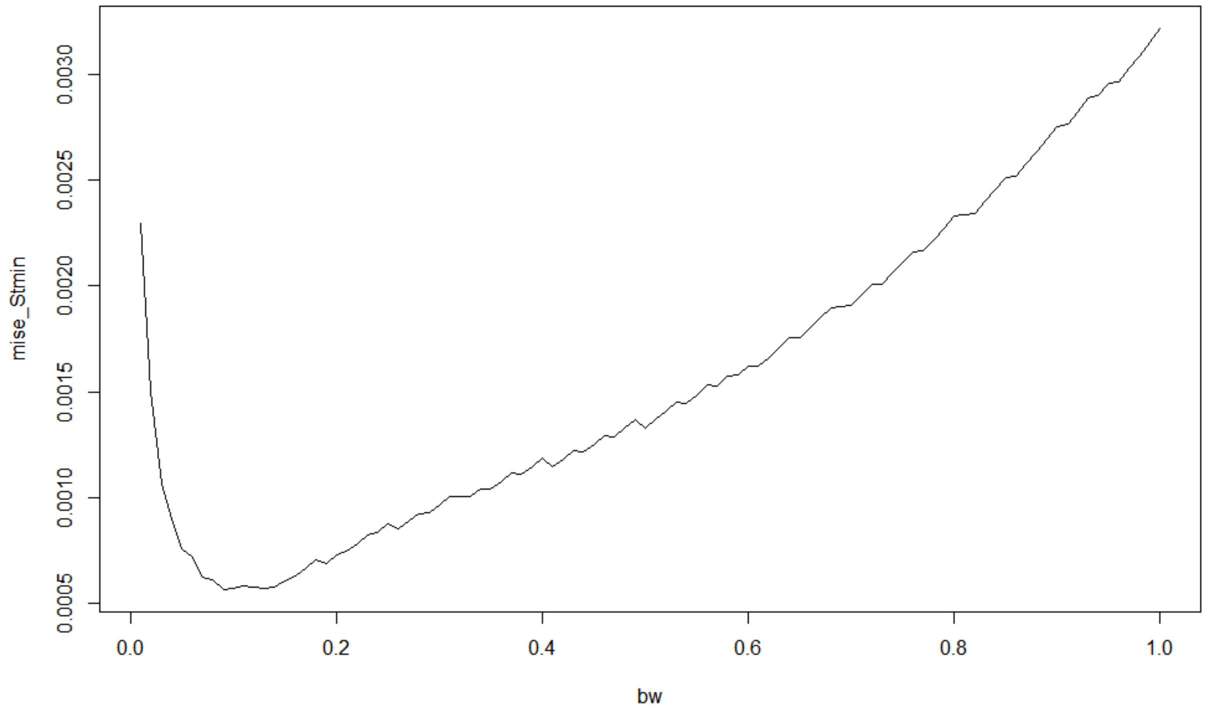

Figure 6. Dependence between h and estimated MISE

```
#vi#
kde<-density(x2, kernel="epanechnikov",
bw=bw[(which(mise_Stmin==min(mise_Stmin)))])
hst<-hist(f_n, breaks="Sturges", col="moccasin", probability=TRUE,
          ylim=c(0,0.75), main="Estimators of density p(x)")
lines(kde, col="mediumaquamarine", lwd=2)
```

```
lines(x, p_x(x), col="mediumorchid", lwd=2)
legend("topleft",legend=c("Epanechnikov kernel", "Sturges rule", "True density"),
       col=c("mediumaquamarine","moccasin","mediumorchid"),
       lty=1:3)
```

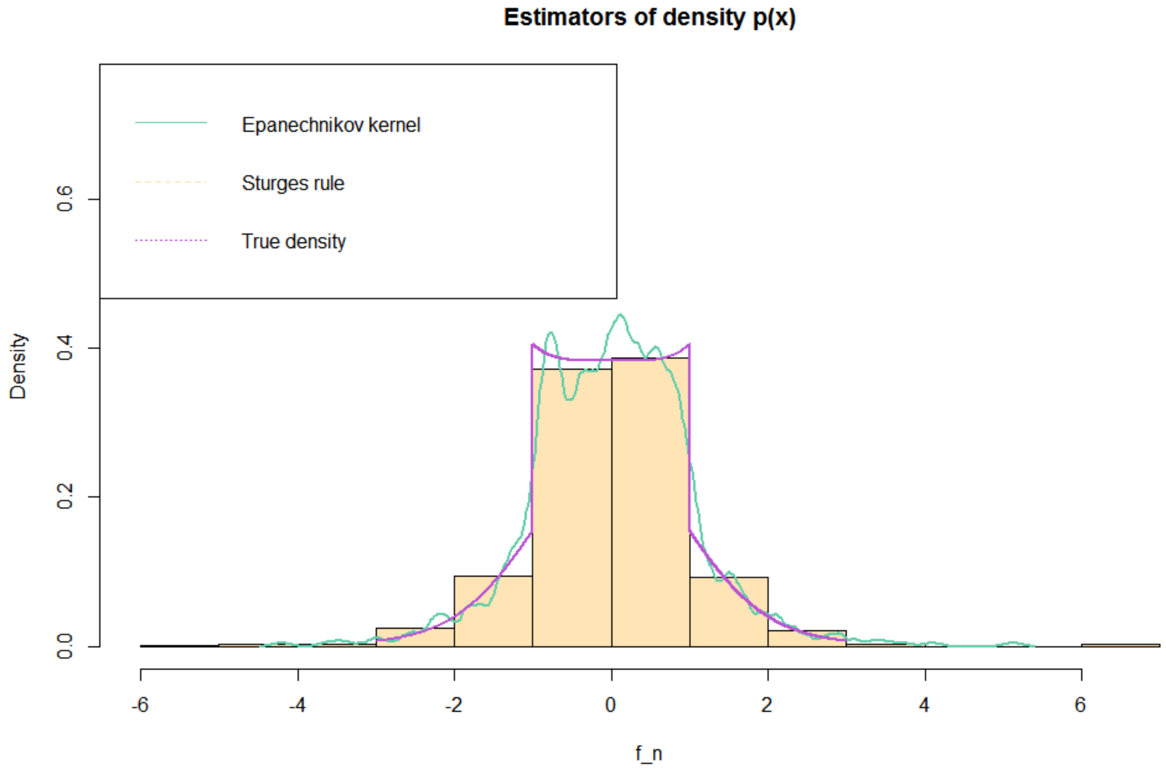

Figure 7. Estimates of density p(x)