(i) Figure 1 (left) shows that $\sigma = .7$ with acceptance rate $\approx .73$.
(ii) Figure 1 (right) shows that $\sigma = 5.9$. with mean squared error $\approx .89$. Simulations using values of $\sigma$ greater than 6 showed sometimes smaller mean squared error, but still with fluctuations.
The R code used is shown below.

![Graphs showing acceptance rate and mean squared error vs. sigma.](image-url)
# Problem 6.21

# target is a normal(0,1)
dtarget <- function(a) dnorm(a, 0, 1)

# define working variables and arrays
nsim <- 10000
nparms <- 60
sigma <- array(0, dim = c(nparms, 1))
accrare <- array(0, dim = c(nparms, 1))
mserror <- array(0, dim = c(nparms, 1))
s <- 0

# generate rvs from candidate Cauchy(0,1)
# they will be reused as Cauchy(0,sigma) by multiplication by sigma

candstandard <- recauchy(nsim-1, 0, 1)

# Cycle through possible values of sigma
for (i in 1:nparms) {
  sigma[i] <- s + .1 * i
dcand <- function(b) dcauchy(b, 0, sigma[i]) # define candidate
candvars <- candstandard * sigma[i] # candidate rvs
rvars <- array(0, dim = c(nsim, 1))
rho <- array(0, dim = c(nsim, 1))
rvars[1] <- rnorm(1, 0, 1) # start in the invariant dist'n
rho[1] <- 0
for (j in 2:nsim) { # loop of Metropolis-Hasting
  cand <- candvars[j-1]
test <- min(((dtarget(cand)/dcand(cand)) * (dcand(rvars[j-1])/dtarget(rvars[j-1]))), 1)
  rho[j] <- runif(1) < test
}
accrare[i] <- sum(rho)/(nsim-1)
mserror[i] <- sum(rvars[2:nsim]^2)/(nsim-1)
}

par(mfrow = c(1, 2))
plot(sigma, accrare, type = "l")
plot(sigma, merror, type = "l")