Week 10 - Solution to exercise

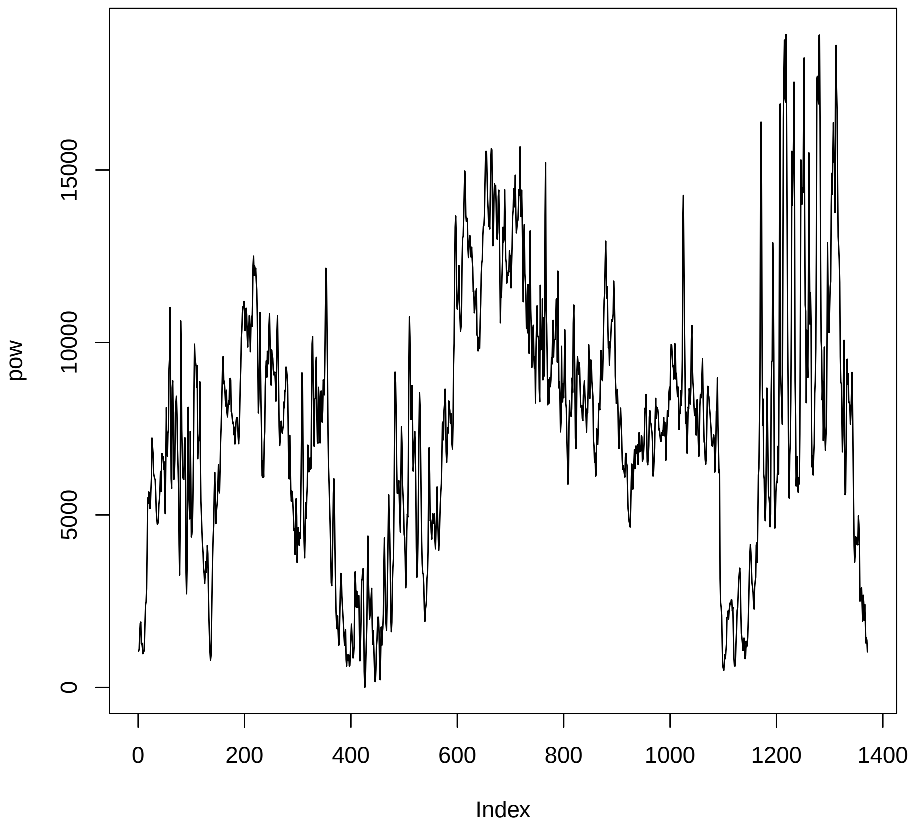

Figure 1: Wind power data.

R code

```
# MWP, 7/6-2011, solution to week 10 exercise
rm(list=ls())

data <- read.table('winddata.txt',header=TRUE)
pow  <- data$pow

### Question a ###
#pdf('windpower.pdf')
plot(pow,type='l')
#dev.off()
```

```
### Question b ###
k <- 3 # AR(k)
resb <- arima(pow,order=c(k,0,0))
print(resb$aic)
#[1] 22804.21

### Question c ##
AR <- function(parvec,x){
  n    <- length(parvec)
  # The order of the AR function is determined by the length of the parvec
  lag  <- n-2
  # Process noise
  sObs <- exp(parvec[1]) # Variance on process noise
  phi  <- parvec[2:n]     # AR parameters
  T    <- length(x)
  l    <- 0
  for (t in 4:T){
    xvec <- c(1,x[(t-lag):(t-1)])
    mu   <- sum(xvec * phi)
    l    <- l + log(dnorm(x[t],mean=mu,sd=sqrt(sObs)))
  }
  -l # Output negative log likelihood value
}

guessc <- c(log(500^2),0,1,0,0)
resc <- nlm(AR,guessc,x=pow,hessian=TRUE,print.level=0)
AICc <- 2*resc$minimum+2*length(guessc)
print(AICc)
#[1] 22751.48
# This value is comparable to the AIC value in question b, although the
# difference is surprisingly large. The reason might be that arima finds
# a local maximum of the likelihood function owing to poor starting values.
# There might also be a slight difference in the implementation of the
# likelihood calculation, in that arima is able to calculate residuals
# for the first three observations even though the model cannot be fitted
# for these values.

### Question d ###
MSAR <- function(parvec,x){
  N    <- length(parvec)
  m    <- 2 # Number of regimes (states in HMM)
  n    <- N/m # Number of parameters in each regime
  lag  <- n-2
  p    <- list(par1=parvec[2:n],par2= parvec[(2+n):(2*n)])
  sObs <- exp(c(parvec[1],parvec[n+1]))

  # The transition matrix is assumed known
  g1    <- 0.95
  g2    <- 0.95
  gamma <- matrix(c(g1,1-g1,1-g2,g2),2)

  T      <- length(x)
  phi    <- matrix(0,T,m)
  l      <- 0
```

```
  delta   <- c(0.5,0.5)
  phi[3,] <- delta# Initial distribution
  likdat  <- numeric(m)
  # Use recursion on page 47 in Zucchini09
  for (t in 4:T){
    # Regime 1
    xvec      <- c(1,x[(t-lag):(t-1)])
    mu        <- sum(xvec * p$par1)
    likdat[1] <- dnorm(x[t],mean=mu,sd=sqrt(sObs[1]))
    # Regime 2
    mu <- sum(xvec * p$par2)
    likdat[2] <- dnorm(x[t],mean=mu,sd=sqrt(sObs[2]))

    v         <- (phi[t-1,] %*% gamma) * likdat
    u         <- sum(v)+1e-20 # Add tiny number to avoid dividing by zero
    l         <- l + log(u)
    phi[t,]   <- v/u
  }
  -l # Output negative log likelihood value
}

guessd <- c(log(900^2),0,0.3,0.3,0.4,log(500^2),0,0.3,0.3,0.4)
resd <- nlm(MSAR,guessd,x=pow,hessian=TRUE,print.level=0)
AICd <- 2*resd$minimum+2*length(guessd)
print(AICd)
#[1] 22176.51
# A clear improvement in the AIC is observed
```