Mikael Simon

3/10/16

time series

HW 3

1.a)

#change alpha level depending on which alpha you want.#

alpha <- -.5

sigma2 <- 1

X <- c(0)

for(t in 2:100){

X[t] <- alpha\*X[t-1] + rnorm(1,sd = sigma2)

}

ar.x<-ar(X)

ar.x

print(summary(ar.x))

predict(ar.x,n.ahead=10)

estimation of parameters:

|  |  |  |
| --- | --- | --- |
| alpha | estimated alpha | estimated sigma^2 |
| 0.9 | 0.895 | 0.9056 |
| 0.5 | 0.5177 | 0.9836 |
| -0.5 | -0.4697 | 1.05 |
| -0.9 | -0.9331 | 1.577 |

predictions for 1 to 10 steps ahead:

alpha= .9

[1] -0.40771043 -0.32412172 -0.24931003 -0.18235374 -0.12242802

[6] -0.06879465 -0.02079290 0.02216854 0.06061894 0.09503195

alpha= .5

[1] -0.78278386 -0.32219134 -0.08376380 0.03965917 0.10354955

[6] 0.13662266 0.15374309 0.16260555 0.16719323 0.16956807

alpha= -.9

[1] -6.148049 5.635397 -5.359302 4.899446 -4.672613 4.258722

[7] -4.074777 3.700903 -3.554297 3.215263

alpha= -.5

[1] -0.29038579 0.19073254 -0.03522583 0.07089605 0.02105566

[6] 0.04446332 0.03346986 0.03863296 0.03620810 0.03734694

1.b)

# change alpha depending on what alpha you want#

alpha<-1.05

x <- vector("numeric",100)

sigma2<-1

x[1:1] <- w[1:1]

for(t in 2:100){{ x[t] <- alpha\*x[t-1]+rnorm(1,sd = sigma2)}

}

ar.x<-ar(x)

ar.x

ts.plot(x)

|  |  |  |
| --- | --- | --- |
| alpha | estimated alpha | estimated sigma^2 |
| 1.01 | 0.97 | 3.208 |
| 1.02 | 0.9484 | 1.603 |
| 1.05 | 0.9461 | 2255 |

2.a)

alpha <- c(5/6,-1/6)

sigma2 <- 1

sample.size <- 1000

alpha.length<-length(alpha)

X <- rnorm(alpha.length)

for(t in (alpha.length+1):sample.size){

X[t] <- alpha[1]\*X[t-1] +

alpha[2]\*X[t-2]+ rnorm(1,sd = sigma2)

}

2.b)

plot(acf(X))



This plot shows that with the time series data there is autocorrelation at different times with the data. The acf also shows that we should use and AR(p) model because it shows a damping sine wave through time.

plot(pacf(X))



This plot shows that the we should use an AR(2) model because there is a spike at lag 2.

2.c)

ar.x<-ar(X, method="yule-walker")

ar.x

#this gives us output of

Coefficients:

1 2

0.7970 -0.1361

Order selected 2 sigma^2 estimated as 0.9997

Shows estimate of alpha1= .7970

Estimate of alpha2=-.1361

Estimate of sigma^2=.9997

And the order of the AR process=2.

ar.x$aic

# also use this command to find the order of the AR model. This command shows the AIC of AR(1) through AR(30) and the model with the lowest AIC is the correct model to use. The output shows that AR(2) has an AIC of zero showing that AR(2) is the correct model to use.

2.d)

c(ar.x$ar[1]-1.96\*sqrt(ar.x$asy.var.coef[1,1]),

ar.x$ar[1]+1.96\*sqrt(ar.x$asy.var.coef[1,1]))

#gives us the 95% confidence interval of the first alpha

gives output of

[1] 0.7355261 0.8585186

Therefore the model parameter of the first alpha=.7970 does fit within the 95% confidence interval.

c(ar.x$ar[2]-1.96\*sqrt(ar.x$asy.var.coef[1,1]),

ar.x$ar[2]+1.96\*sqrt(ar.x$asy.var.coef[1,1]))

#gives us the 95% confidence interval of the second alpha

gives output of

[1] -0.19759195 -0.07459941

Therefore the model parameter of the second alpha=.-.1361 does fit within the 95% confidence interval.

2.f)

ar.x.resid<-ar.x$resid[3:1000]

acf(ar.x.resid)



This plot shows that there is no autocorrelation throughout time for the residuals of the fitted model because there is no significant spikes in lag, except for lag=0.