Stat 461 HW4

Roshan Poduval

11/17/2020

3b

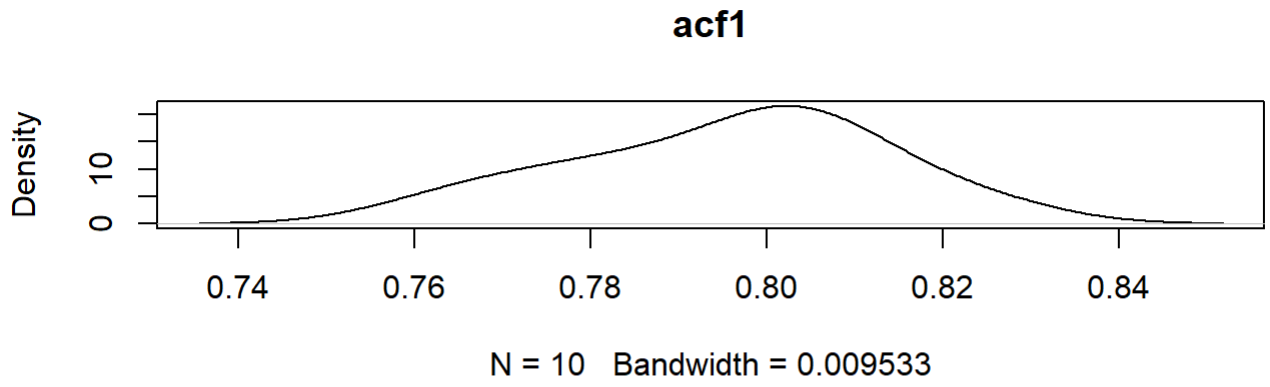
```
set.seed(1234)
T = 1000
N = 10000
Y = matrix(, nrow=1000, ncol=N/T)
for (i in 1:(N/T)) {
  Y[,i] = arima.sim(model=list(ar=c(0.8)), n=T)
}
print(acf(Y[,1], lag=5, main="", plot=FALSE)[c(1,5)])
```

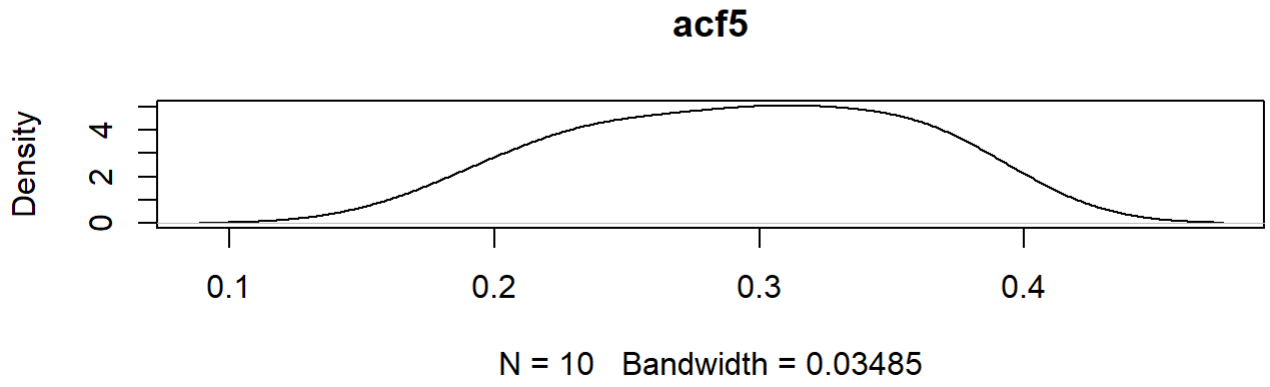
```
##
## Autocorrelations of series 'Y[, 1]', by lag
##
##     1     5
## 0.801 0.298
```

```
my_acfs = matrix(, nrow=2, ncol=N/T)
for (i in 1:(N/T)) {
  my_acfs[1,i] = acf(Y[,i], lag=5, main="", plot=FALSE)[1]$acf
  my_acfs[2,i] = acf(Y[,i], lag=5, main="", plot=FALSE)[5]$acf
}
```

3e

```
acf1 = my_acfs[1,]
acf2 = my_acfs[2,]
par(mfrow=c(2,1))
plot(density(acf1), main="acf1")
plot(density(acf2), main="acf5")
```






```
paste("acf1; mean:", mean(acf1), "sd:", sd(acf1))
```

```
## [1] "acf1; mean: 0.795480109270835 sd: 0.0177323385836998"
```

```
paste("acf5; mean:", mean(acf2), "sd:", sd(acf2))
```

```
## [1] "acf5; mean: 0.292728570789667 sd: 0.0613727922762759"
```

4e

```
ten_predict_error <- function(a110, a109, a108) {
  return(a110 - (.3 * a109) + (.12 * a108))
}

n_samples = 1000

a110 = rt(n_samples, 4)
a109 = rt(n_samples, 4)
a108 = rt(n_samples, 4)

predictions = ten_predict_error(a110, a109, a108)

a = mean(predictions)
s = sd(predictions)

error = qnorm(.975)*s/sqrt(n_samples)
paste0("95% C.I. of 10-step ahead prediction error(n=", n_samples,"): (", a-error, ", ", a+erro
r, ")")
```

```
## [1] "95% C.I. of 10-step ahead prediction error(n=1000): (-0.109011459873612, 0.0866523929473
549)"
```