stat153hw

Shicheng Huang

October 29, 2016

Q1

```
set.seed(1)
phis <- c(0.2, 0.6, 0.95)
sample_sizes <- c(20, 60, 100)
for (phi in phis) {
  for (n in sample_sizes) {
    #sim_phi(phi, n)
  }
}
## update to graphical representation later
```

The estimated sample mean tend to be smaller than the actual sample mean. But as the sample size increases, the estimated sample mean gets closer to the actual sample mean. The sample variance has been pretty accurate for 0.2 and 0.6 regardless of sample size but not for 0.95.

Q2

```
thetas <- list(first = c(1.5, 0.75), second = c(0.95, 0))
for (i in 1:length(thetas)) {
  for (n in sample_sizes){
    temp_theta = thetas[[i]]
    #sim_theta(temp_theta[1], temp_theta[2], n)
  }
}
```

The standard covariance estimate error is within 0.02(good) when sample size is at 60 and 100. The first case one parameter gets more accurate at the cost of the other as the sample size increase. I think maybe because the theta1 is greater than 1; in the second case, both parameters benefit from the increase in sample size.

One things to note is that I run into optimization warning when estimating thetas using the mle method even though I have increased the maximum iteration.

Q3

```
data("LakeHuron")
#(a) Fit a linear trend function to the data and obtain residuals.
lmod <- lm(LakeHuron~time(LakeHuron))
#(b) Fit an AR(1) model to the residuals using the R function arima()
res_mod <- arima(x = lmod$residuals, order = c(1, 0, 0))
# (c) Obtain predictions for the residuals for the future m = 30 time points without using the
# predict function in R.
```

```r
ephi <- res_mod$coef[1]
emu <- res_mod$coef[2]
res_pred1 <- unname(ephi * (lmod$residuals[length(LakeHuron)] - emu) + emu)
for (i in 2:30) {
  res_pred1[i] <- ephi * (res_pred1[i - 1] - emu) + emu
}
# (d) Compare your predictions with those obtained by the predict function.
res_pred2 <- predict(res_mod, n.ahead = 30)$pred
res_pred1 == res_pred1
```

```
##  [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [15] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [29] TRUE TRUE
```

```r
# they are the same.

# (e) Obtain predictions for the original data for the future m = 30 time points.
linear_pred <- lmod$coefficients[1] + lmod$coefficients[2] * (1973 : 2002)
data_pred <- linear_pred + res_pred1
```

Q4

(a) Is it a good idea to fit the AR(2) model to this dataset? Why or why not?
Yes, because the pacf cuts off(within confidence interval) at lag2.

(b) -0.6. Since the pacf with lag2 is approximately -0.6 from the graph. And pacf lag(n) is cofficient for ar(p) model.

Q5

Because the ar components also influnences the model, or can be counted as part of the ma component. In fact, we can revert the estimated model to a model that is close to our original model by reducing redundent parameter.

$$\phi(z)X_t = \theta(z)W_t$$
$$(1 - 0.7838z)X_t = (1 - 0.08z - 0.53z^2)W_t$$
(*by polynomial division we have approximately*)
$$X_t \approx (0.964778 + 0.676193z)W_t$$
$$X_t \approx (1 + 0.676z)W_t$$

As we can see our estimated model could be simplied as a MA(1) model with parameter 0.676, which is close to our data generating model.So the R result makes sense.

Q6

a, MA(7): No, Because the autocorrelation is not decreasing from lag1 to 7, instead, it cuts off starting from lag2 but spikes at lag5, 6 and 7.
b, MA(1) with period of 6: No, the pacf doesn't tails off before lag6. c, MA(1) with differencing 1, period 6. I think it is reasonable. I think it needs differencing because the pacf is too high from lag1 to lag5

Q7

Q8

(a) The first model is a random walk, so it is expected to be wondering about 0, much higher flucutations(change of ups and downs);Because X_t is linearly dependent on t, and a 0-mean noise variable at time t, the second model will look like a straight line or cluster like a straight line.

(b) No. the variance of X_t is the sum of t i.i.d random noise, $t\sigma_z$, so it is not independent of time.
?(c) X_n, the conditioned(knowing all the past values) random walk is a martingale.

Q9

(a)

(b)

(c)

(d)

(e)

Q10

(a) Show that $X_t = Y_t - Y_{t-d}$ is a stationary process

$$\begin{aligned} X_t &= Y_t - Y_{t-d} \\ &= s_t + Z_t - s_{t-d} - Z_{t-d-1} \end{aligned}$$

(b)