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)
  }
}
```

```
## Theoretical Sample Distribution
## size: 20 phi: 0.2 se: 0.048
## Estimated Sample Distribution:
## phi: 0.114 se: 0.0427
##
## Theoretical Sample Distribution
## size: 60 phi: 0.2 se: 0.016
## Estimated Sample Distribution:
## phi: 0.168 se: 0.0153
##
## Theoretical Sample Distribution
## size: 100 phi: 0.2 se: 0.0096
## Estimated Sample Distribution:
## phi: 0.183 se: 0.0094
##
## Theoretical Sample Distribution
## size: 20 phi: 0.6 se: 0.032
## Estimated Sample Distribution:
## phi: 0.428 se: 0.0377
##
## Theoretical Sample Distribution
## size: 60 phi: 0.6 se: 0.01066667
## Estimated Sample Distribution:
## phi: 0.541 se: 0.012
##
## Theoretical Sample Distribution
## size: 100 phi: 0.6 se: 0.0064
## Estimated Sample Distribution:
## phi: 0.566 se: 0.007
##
## Theoretical Sample Distribution
## size: 20 phi: 0.95 se: 0.004875
## Estimated Sample Distribution:
## phi: 0.657 se: 0.0281
##
## Theoretical Sample Distribution
## size: 60 phi: 0.95 se: 0.001625
## Estimated Sample Distribution:
## phi: 0.853 se: 0.0055
##
## Theoretical Sample Distribution
## size: 100 phi: 0.95 se: 0.000975
## Estimated Sample Distribution:
## phi: 0.893 se: 0.0026
```

```
## 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

```
set.seed(2)
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)
  }
}
```

```
## sample size: 20 theta1,theta2: 1.5 0.75
## The estimated means of theta1 and theta2 are 1.527 0.728
## The theoretical covariance between theta1 and theta2 are 0.13125
## The estimated covariance between theta1 and theta2 are 0.085
##
## sample size: 60 theta1,theta2: 1.5 0.75
## The estimated means of theta1 and theta2 are 1.455 0.671
## The theoretical covariance between theta1 and theta2 are 0.04375
## The estimated covariance between theta1 and theta2 are 0.0485
##
## sample size: 100 theta1,theta2: 1.5 0.75
## The estimated means of theta1 and theta2 are 1.453 0.678
## The theoretical covariance between theta1 and theta2 are 0.02625
## The estimated covariance between theta1 and theta2 are 0.0357
```

```
## sample size: 20 theta1,theta2: 0.95 0
## The estimated means of theta1 and theta2 are 0.997 0.111
## The theoretical covariance between theta1 and theta2 are 0.0475
## The estimated covariance between theta1 and theta2 are 0.112
##
## sample size: 60 theta1,theta2: 0.95 0
## The estimated means of theta1 and theta2 are 0.98 0.025
## The theoretical covariance between theta1 and theta2 are 0.01583333
## The estimated covariance between theta1 and theta2 are 0.0206
##
## sample size: 100 theta1,theta2: 0.95 0
## The estimated means of theta1 and theta2 are 0.966 0.009
## The theoretical covariance between theta1 and theta2 are 0.0095
## The estimated covariance between theta1 and theta2 are 0.0118
```

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.

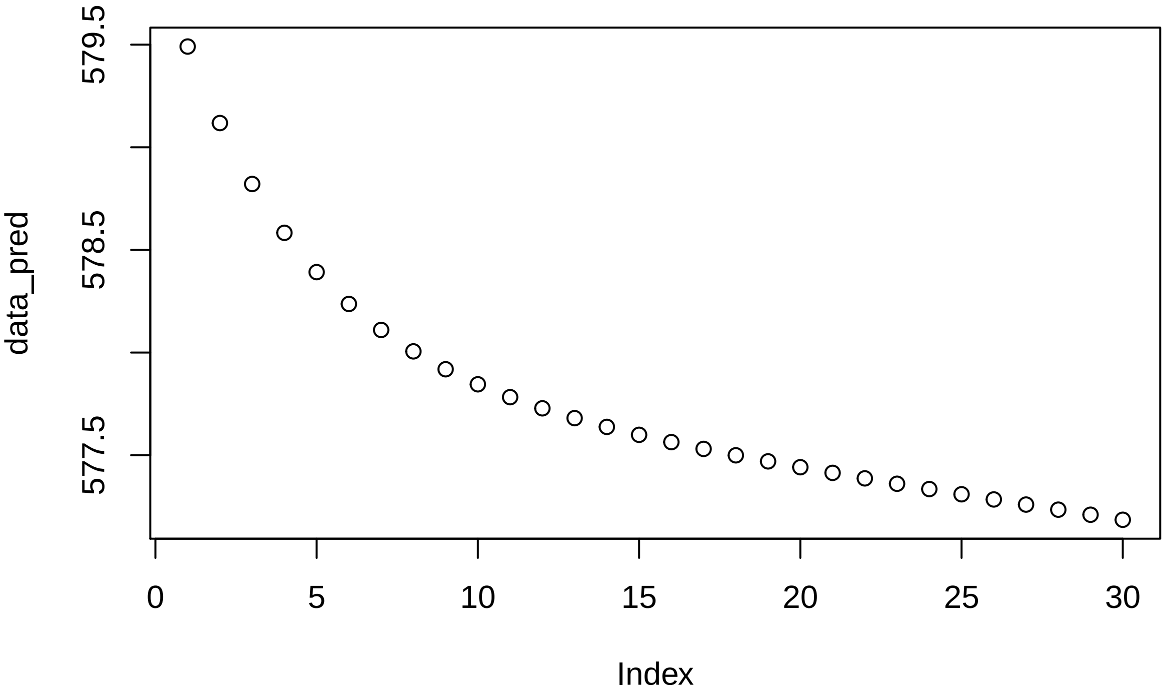
Q3

```
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$residuals, order = c(1, 0, 0))
# (c) Obtain predictions for the residuals for the future m = 30 time points without using the
# predict function in R.
ephi <- res_mod$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_pred2
```

```
## Time Series:
## Start = 99
## End = 128
## Frequency = 1
##  [1]  TRUE  TRUE  TRUE FALSE  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE
## [12]  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE  TRUE
## [23]  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE
```

Although numerically res_pred1 and res_pred2 are the same but R has some “weird” way of comparing “numbers” so not all the values are the “same” in R.

```
# (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
plot(data_pred)
```



Q4

(a) 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.

$$\begin{aligned}
\phi(z)X_t &= \theta(z)W_t \\
(1-0.7838z)X_t &= (1-0.08z-0.53z^2)W_t \\
(\textit{by polynomial division we have approximately}) \\
X_t &\approx (0.964778+0.676193z)W_t \\
X_t &\approx (1+0.676z)W_t
\end{aligned}$$

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 and the absolute value is declining along the lags.

Q7

let X_t denote the original time series data, ${}_sX_t$ as the time series after differencing X_t by period of 12. Y_t the time series after differencing ${}_sX_t$ by lag 1.

$$\begin{aligned} Y_t &= {}_sX_t - {}_sX_{t-1} \\ &= X_t - X_{t-12} - (X_t - X_{t-12-1}) \\ &= X_t - X_{t-12} - X_{t-1} + X_{t-13} \end{aligned}$$

thus by applying shifting the equation forward and solve for X_{t+1} we have

$$\begin{aligned} X_{t+1} &= X_t + X_{t-11} - X_{t-12} + \hat{Y_{t+1}} \\ &= 381.62 + 382.44 - 379.00 + (-1.168874) \\ &\approx 382.90 \end{aligned}$$

Q8

(a) Dataset two is the first model, also a random walk, because the model is expected to be wondering about 0, much higher variance across time(change of ups and downs);Dataset one is the second model: 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 close to 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)

$$\begin{aligned} X_{t+1} &= \delta X_{t+1} + X_t \\ &= Z_{t+1} + X_t \end{aligned}$$

So predicting a random walk is like predicting the next value of a white noise process, given the present value. Therefore the best one step forecast should be the avg value of the white noise process, which is 0, plus the present value X_t, then we get X_t; we can also arrive at the same conclude by realizing a random walk is a martingale process.

Q9

(a) $\phi(z) = 1$ and $\theta(z) = 1 + \frac{1}{2}z$
Because the acf cuts off at 1, I model it with MA(1) model. get the cofficient of $\theta_1 using$ $acf(1) = \frac{\theta_1^2}{1-\theta_1^2} = 0.4$.

(b) $\phi(z) = 1$ and $\theta(z) = 1 + \frac{z^3}{2}$
Because the acf only high at 3(besides 0), and not 2, I mode it with MA(1) with period d.

(c) $\phi(z) = 1 + 0.5z$ and $\theta(z) = 1$
Modeling it with AR(1) because for AR(1) $\frac{\rho_X(h+1)}{\rho_X(h)} = \frac{\phi^{h+1}}{phi^h} = \phi$

(d) $\phi(z) = 1 + 0.5z$ annd $\theta(z) = 0.3$
Modeling it with ARMA(1, 1) because the AR(1) compoment makes the fraction between $\rho(h+1)$ and $\rho(h)$ equals to ρ and the MA(1) makes $\rho_X(1) \neq 0.5$

(e) $X_t = W_t + 0.5W_{t-12}$. To create a "fake"-seasonal behavior I use the MA(12) models with cofficients 0 for lag 1 to 11. It is stationary because $E[X_t] = 0$. Variance and covariance are just variance of noises and depends only on the period and lag.

Q10

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

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

It is stationary bc its expectation is 0 and variance σ_Z^2. The autocovariance check see section (c).

(b) Show that the best linear predictor for X_{nd+1} in terms of $X_1, X_2, ...X_{nd-1}, X_{nd}$ equals $\frac{-X_1 - 2X_{1+d} - 3X_{1+2d} - ... - nX_{1+(n-1)d}}{n+1}$

** EC omitted**

(c) Find the partial autocorrelation function of X_t

Since $X_t = Z_t - Z_{t-d}$, X_t could be regarded as an $MA(1)_d$ process. So the pacf for non-period lags should be 0, and the lags that are divisible by period d should sort of follow the pacf of an MA(1) process, defined as $-\frac{(-\theta)^h}{1+\theta^2+...\theta^{2h}}$. Since through simulation I know all periodic cofficients should be negative, therefore that previous formua could be motified as $-\frac{1}{1+\theta^2+...\theta^{2h}} = -\frac{1}{1+h}$

$$\phi_{hh} = \begin{cases} -\frac{1}{1+m} & when\ md = h, m \in \mathbb{N} \\ 0 \quad when\ md = h, m \notin \mathbb{N} \end{cases}$$

Q11

Determine whether each of the following statements is true or false. Provide reasons in each case. You will not be awarded points if no reason is provided

(a) The sample autocorrelations of data coming from a white noise process can themselves be treated as white noise.

T.Because the sample autocorrelations are bounded by 1 and -1 with expectations 0 because white noise are indepedent by definition.

(b) The correlations outputed by the ARMAacf function in R can be larger than one if the ARMA parameters correspond to a non-causal process.

T.Because the function assumes $\sum_{j=0}^{\infty} \psi_j < \infty$, then the theoretical acf bounds by [1, 1]. Non-causal process doesn't statisfy this condition so the acf($\frac{\sum_{j=0}^{\infty} \psi_j \psi_{j+h}}{\sum_{j=0}^{\infty} \psi_j}$) is greater than one for some h.

(c) If a process is weakly stationary and gaussian, then it is strongly stationary. **False.This is because a multivariate Gaussian distribution is fully characterized by its first two moments.**

(d) For every stationary process, the partial autocorrelation at lag h is always less than or equal to the autocorrelation at lag h.

By definition, the partial autocorrelation at lag h is after we subtract the linear dependence between X_h and X(we are making them less dependent), so partial autocorrelation is always smaller.

(e) The exponential smoothing method for forecasting does not require any specification of tuning parameters.

No we have to tune the weight between past forecast and actual values at time t.