Personal Savings Analysis - Rough Draft

Andrew Brown, Melissa Hooke, Frances Hung, Mai Nguyen, Brenner Ryan

11/29/2018

Time Series Exploration

```r
# load the time series
savings = read.csv("savings.csv",header=TRUE, nrows=104)
savings = savings %>% select(2)
savings.entire = savings[1:84,]
savings<-ts(savings, start=c(1955),frequency=4)

# plot the original time series
plot(savings, xlab="Year (by Quarter)",
     ylab= "Percent of Disposible Income",
     main= "Time Series of Personal Savings in the US (1955-1980)")
points(y=savings,x=as.vector(time(savings)),pch=as.vector(season(savings)), cex=.75)
abline(mean(savings),0)
```
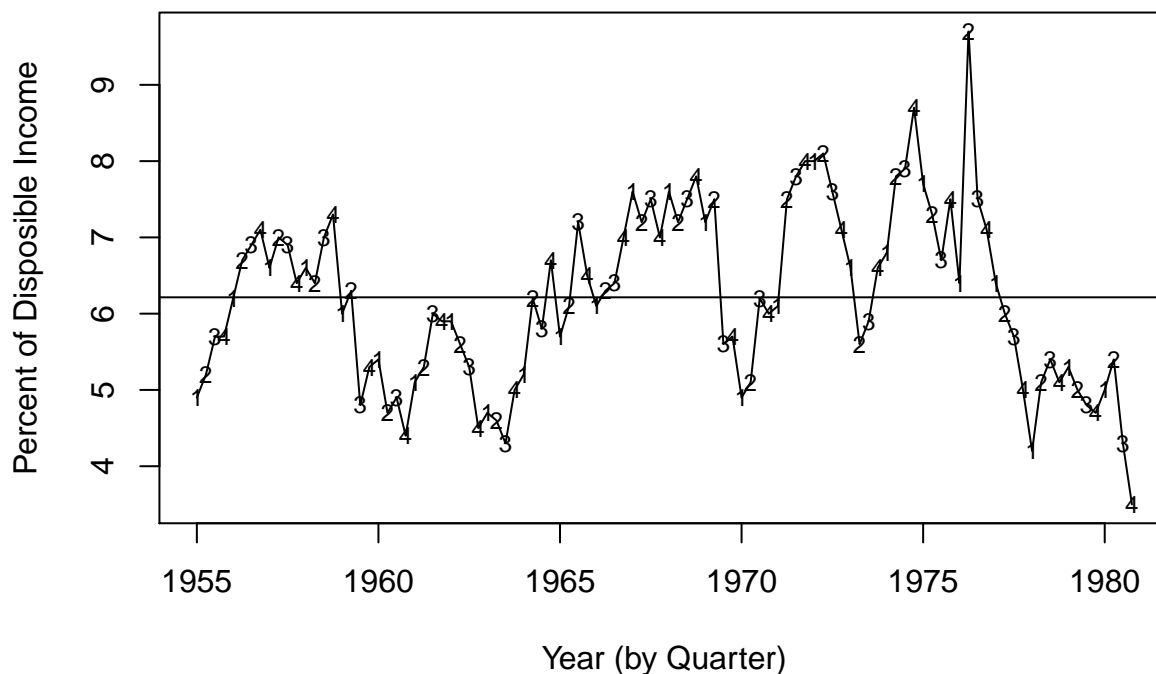

The time series spans over 26 years with each year divided into financial quarters, amounting to 104 total observations. From our original time series plot, we do not see any obvious seasonal trends. However, there seems to be a positive correlation between terms close to one another. The general increasing trend and

volatility in the 70s may be linked to the economic crash in the early 70s and oil energy crisis in 1979. The 1970's were marked by high inflation and growing expenses due to rising interest rates. We took off the last 5 years because there was a huge drop due to a recession. Then we took off the last 6 observations in order to use them as test points at the end. The resulting series is below:

```
# set aside points to validate our forecasts
savings.test = savings[79:84,]

# remove the last 5 years because of the huge dip due to recession
savings = savings[1:78,]
savings<-ts(savings, start=c(1955),frequency=4)

# plot the shortened time series with labels for quarters
plot(savings, xlab="Year (by Quarter)",
     ylab= "Percent of Disposible Income",
     main= "Time Series of Personal Savings in the US (1955-1980)")
points(y=savings,x=as.vector(time(savings)),pch=as.vector(season(savings)), cex=.75)
```

## Time Series of Personal Savings in the US (1955–1980)
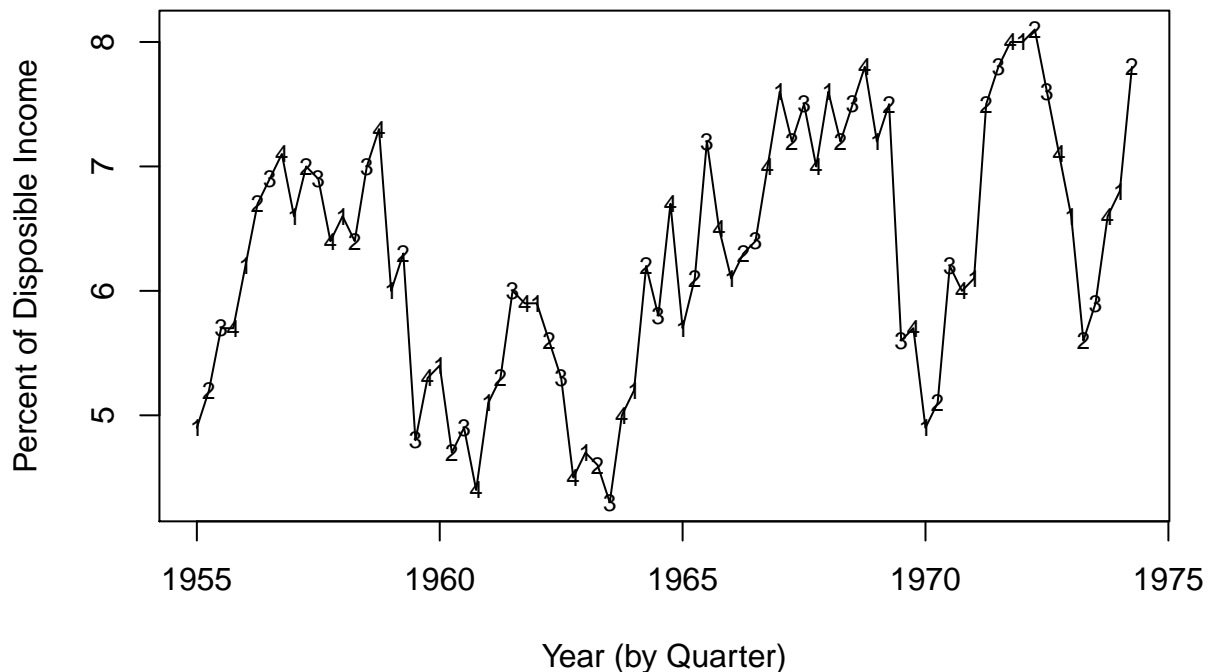
There is a general upward trend in the data, as well as a dependence on previous quarters' savings to determine current personal savings percentages. We may want to take a difference to see if that helps with stationarity, but first lets explore without taking the difference.

```
# should we do a transformation?
boxcox = BoxCox.ar(savings)
boxcox$mle
```
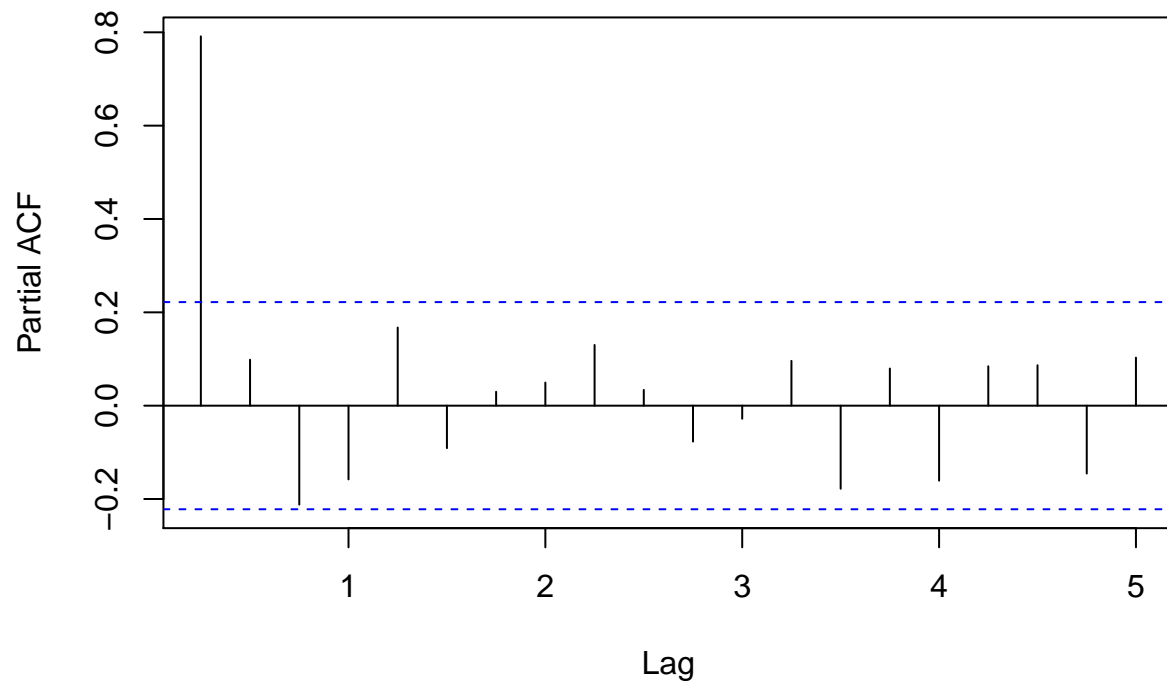
The Boxcox output indicates that a transformation is not necessary in order to stabilize the variance since λ is about equal to 1.

```r
acf(savings, lag.max = 20)
```
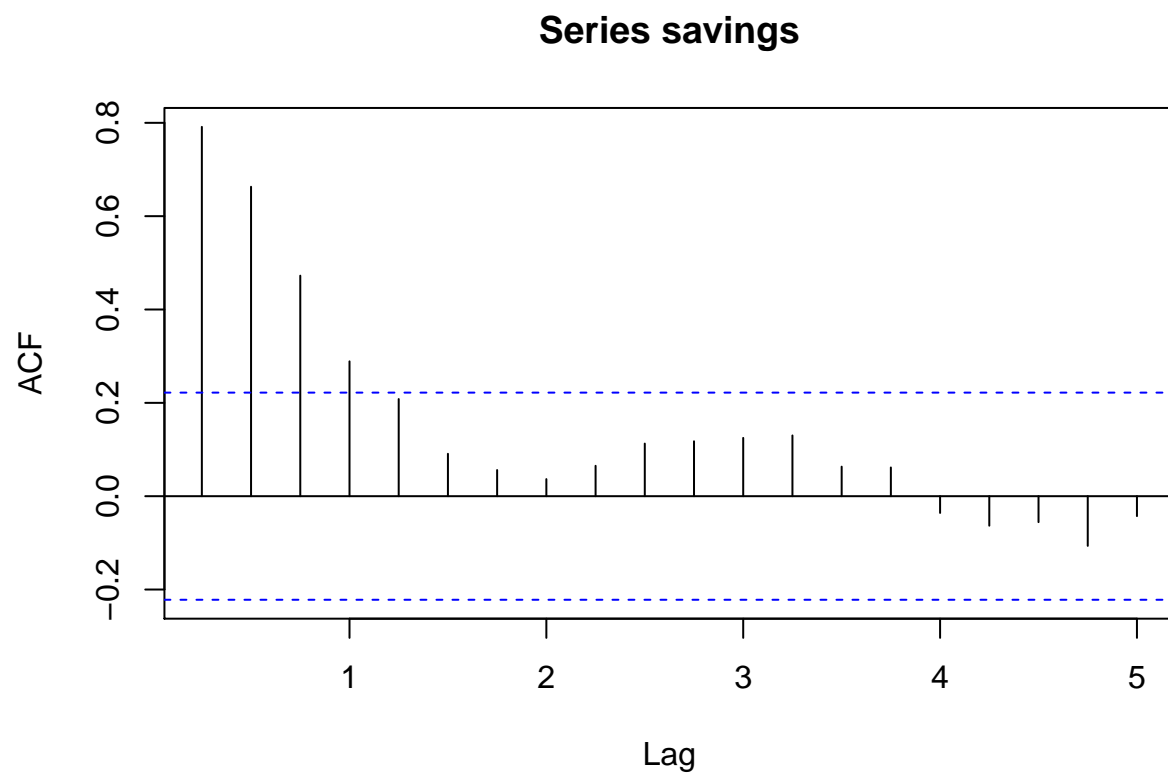
Series savings


```r
pacf(savings, lag.max = 20)
```
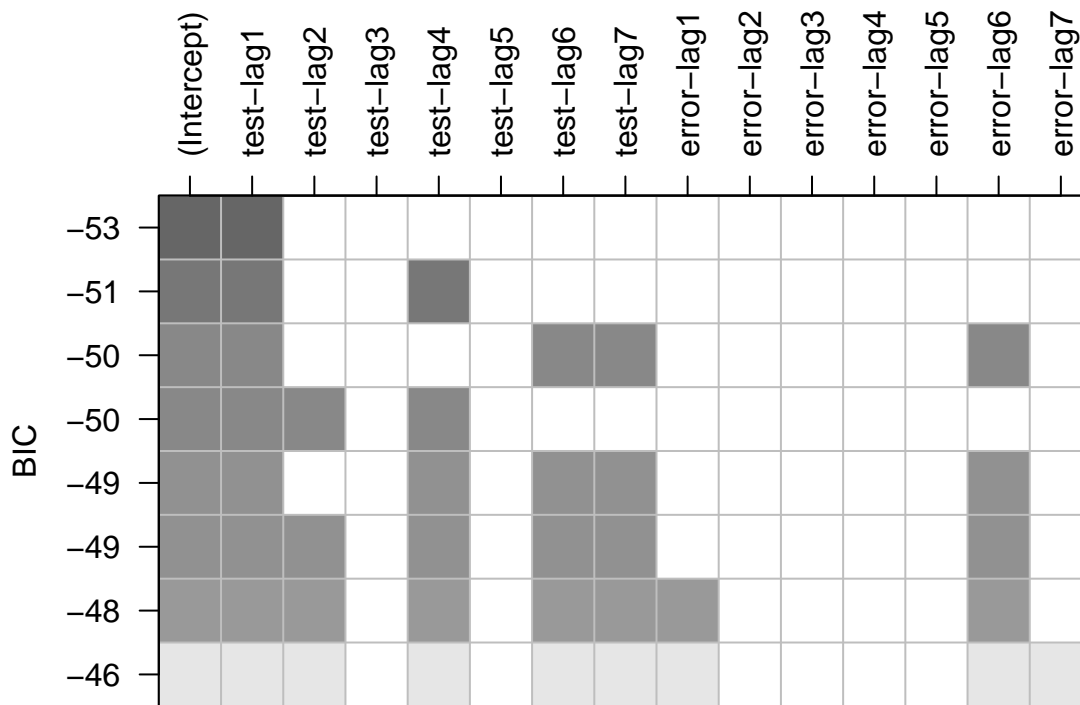
Series savings

The PACF seems to indicate that an AR(1) process may be a good candidate model.

Model Specification

```
# use the eacf and best subsets to find a candidate model
eacf(savings)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x x o o o o o o o  o  o  o
## 1 o x o o o o o o o o o  o  o  o
## 2 x x o o o o o o o o o  o  o  o
## 3 x x o o o o o o o o o  o  o  o
## 4 x o o o o o o o o o o  o  o  o
## 5 x x o o o o o x o o o  o  o  o
## 6 x o o o o o o o o o o  o  o  o
## 7 x o o o o o o o o o o  o  o  o
```

```
sub = armasubsets(y=savings,nar=7,nma=7, y.name='test', ar.method='ols')
plot(sub)
```

BIC plot axis labels (columns): (Intercept), test-lag1, test-lag2, test-lag3, test-lag4, test-lag5, test-lag6, test-lag7, error-lag1, error-lag2, error-lag3, error-lag4, error-lag5, error-lag6, error-lag7
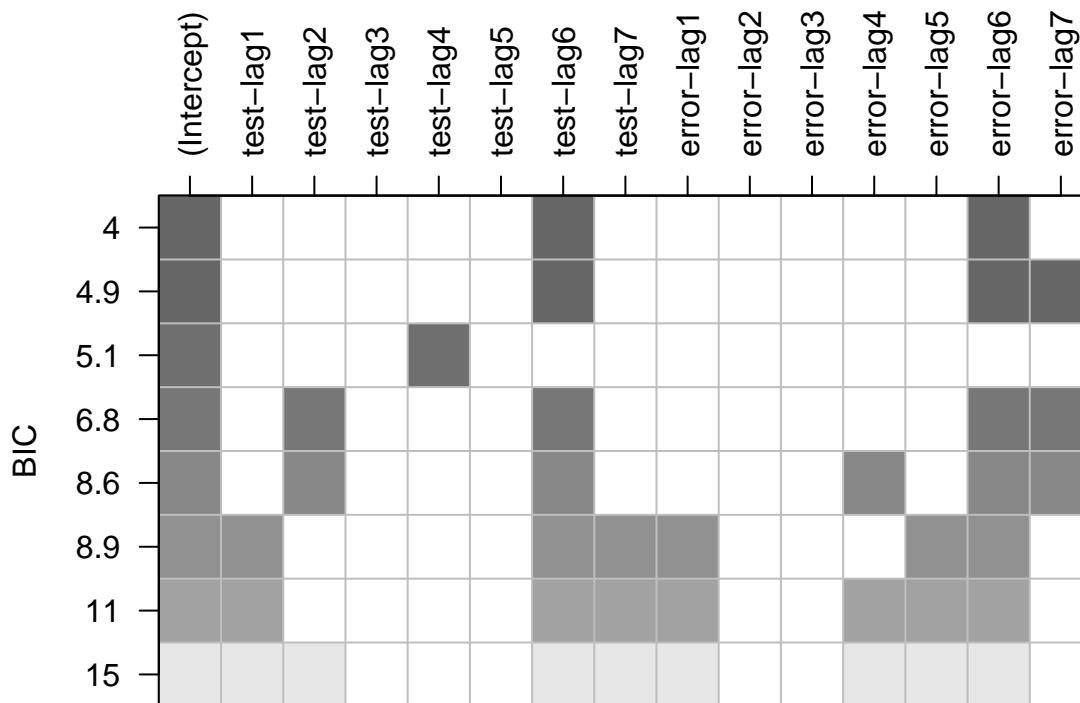
BIC values (rows): −53, −51, −50, −50, −49, −49, −48, −46

While the EACF is inconclusive, the best subsets method indicates that, an AR(1) process or a multiplicative $AR(1) \times AR(1)_4$ with a seasonal period of 4 are good candidate models.

```
diffs = (savings-zlag(savings))[2:78]
eacf(diffs)
```

```
## AR/MA
##    0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 o o o o x o x o o o o  o  x  o
## 1 x o o x o o o o o o o  o  o  o
## 2 x x o x o o o o o o o  o  o  o
## 3 x x o o o o o o o o o  o  o  o
## 4 o x o o o o o o x o o  o  o  o
## 5 o x o o o o o o o o o  o  o  o
## 6 x x o o o o o o o o o  o  o  o
## 7 x o x o o o o o o o o  o  o  o
```

```
sub = armasubsets(y=diffs,nar=7,nma=7, y.name='test', ar.method='ols')
plot(sub)
```

BIC

(Intercept) | test–lag1 | test–lag2 | test–lag3 | test–lag4 | test–lag5 | test–lag6 | test–lag7 | error–lag1 | error–lag2 | error–lag3 | error–lag4 | error–lag5 | error–lag6 | error–lag7

4 — 4.9 — 5.1 — 6.8 — 8.6 — 8.9 — 11 — 15

For the differenced series, a seasonal period of 6 is being suggested, which doesn't seem to make much sense.

Model Fitting

```r
# fit an AR(1) process
arima(savings, order = c(1, 0, 0), seasonal = list(order = c(0, 0, 0)), method=c('ML'))
```

```
##
## Call:
## arima(x = savings, order = c(1, 0, 0), seasonal = list(order = c(0, 0, 0)),
##     method = c("ML"))
##
## Coefficients:
##          ar1  intercept
##       0.8250     6.2768
## s.e.  0.0652     0.3543
##
## sigma^2 estimated as 0.336:  log likelihood = -68.71,  aic = 141.43
```

Here are our parameters for the AR(1) model. Given these parameters, the AR(1) model is $Y_t - 6.28 = .825(Y_{t-1} - 6.28) + e_t$.

```r
# fit the seasonal model
arima(savings, order = c(1, 0, 0), seasonal = list(order = c(1, 0, 0), period = 4), method=c('ML'))
```

```
##
```

```
## Call:
## arima(x = savings, order = c(1, 0, 0), seasonal = list(order = c(1, 0, 0), period = 4),
##     method = c("ML"))
##
## Coefficients:
##          ar1      sar1  intercept
##       0.8610  -0.2276     6.2710
## s.e.  0.0596   0.1157     0.3514
##
## sigma^2 estimated as 0.3198:  log likelihood = -66.87,  aic = 139.75
```

And here are the parameters for the multiplicative seasonal model. Given these parameters, the seasonal model is given by $(Y_t - 6.27)(1 - .861(B - 6.27))(1 + .228(B - 6.27)^4) = e_t$

Diagnostics

These two models are almost indistinguishable in terms of error. The AR(1) model has a higher AIC and BIC, but the seasonal model has a standard error, and none of the differences between the two models are substantial. Our error measurements don't give us much insight into which model is best, so we should turn to forecasting to see which model is more accurate in predicting future terms.

Forecasting

```r
# forecasts for the AR(1) process
set.seed(47)
forecasts = c()
Y_t1 = savings[78]
for (i in 1:6) {
  Y_t = .825*(Y_t1-6.28) + rnorm(1,0,sqrt(.336)) + 6.28
  forecasts[i] = Y_t
  Y_t1 = Y_t
}
```

```r
# plot the forecasts
plot(savings.entire, type="o")
points(79:84,forecasts, col="red")
```
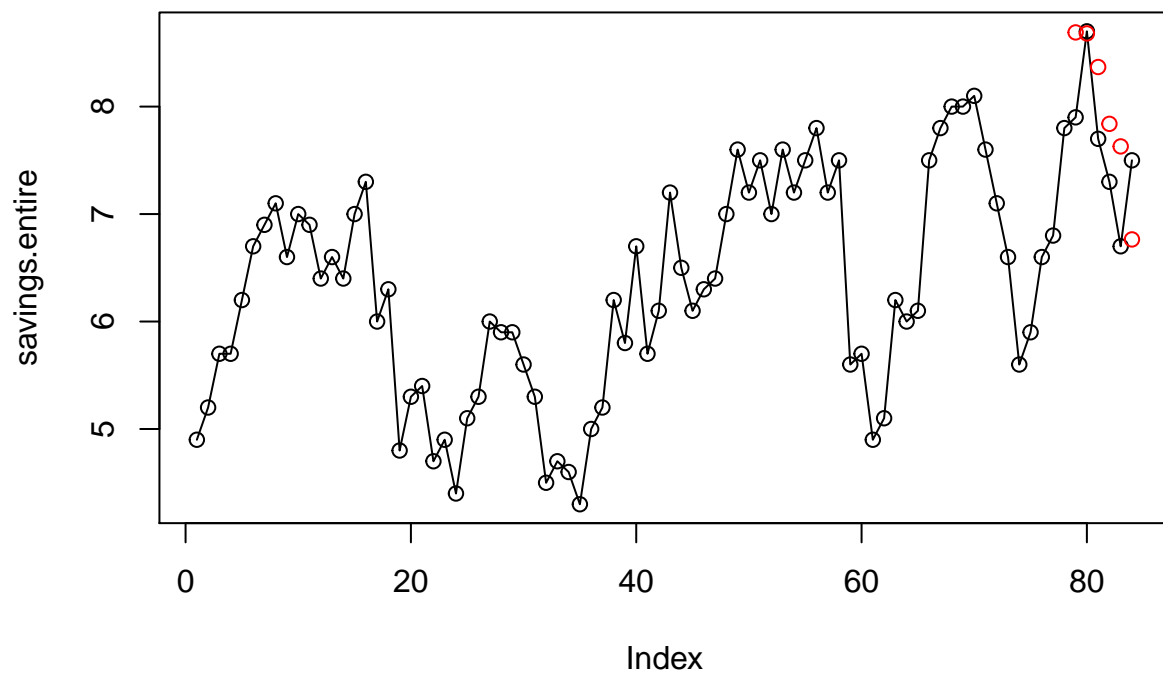
The forecasts from the AR(1) model are pretty good!