Econ144_hw4

Sijia Hua

5/18/2019

Data set up

I download the most recent data from Freddie Mac House, and organized it.

```
setwd("/Users/Renaissance/Desktop")
data_new <- read_xls("11.xls")
```

```
## New names:
## * `` -> ...8
## * `` -> ...9
```

```
# read the most recent data set
data_recent <- read_xls("MSAs_SA.xls")
```

```
## New names:
## * `` -> ...1
## * `Economic and Housing Research` -> `Economic and Housing Research...2`
## * `` -> ...3
## * `` -> ...4
## * `` -> ...5
## * ... and 210 more problems
```

```
data_recent <- data_recent[c(-1:-4),]
# data for Albany
data_a <- data_recent[c(6)]
data_recent2 <- read_xls("MSAs_SA.xls",sheet = 2)
```

```
## New names:
## * `` -> ...1
## * `Economic and Housing Research` -> `Economic and Housing Research...2`
## * `` -> ...3
## * `` -> ...4
## * `` -> ...5
## * ... and 164 more problems
```

```
data_recent2 <- data_recent2[c(-1:-4),]
# data for SF and SJ
data_ss <- data_recent2[c(98,99)]
# combine data of AS,SF,SJ
data_three <- cbind(data_recent[c(1)],data_a,data_ss)
data_three <- na.omit(data_three)
# construct a data frame of quarterly data
data_three2 <- data.frame()
data_three2 <- c(data_three[1,])

## convert new monthly data into quarterly by choosing tthe data of one quarter's last month
for (i in 1:177)
{
  data_three2<- rbind (data_three2,data_three[1+3*i,])
}
```

11.1

descrption: This is the Freddie Mac House Price Index data obtained from Freddie Mac. It is a measure of typical price inflation for houses within the United States. The data begins in 1975 and collects each month. For 11.1, I choose Santa Jose and San Franciso to do the comparision.

```
# create a function for finding growth rate
findg<-function(data)
{
  g <- (as.numeric(data)/Lag(as.numeric(data),1)-1)*100
  g <- na.omit(g)
}
# construct a data frame of growth rate in these three area
gr <- apply(data_three2,2,findg)
```

```
## Warning in FUN(newX[, i], ...): NAs introduced by coercion
## Warning in Lag(as.numeric(data), 1): NAs introduced by coercion
## Warning in FUN(newX[, i], ...): NAs introduced by coercion
## Warning in Lag(as.numeric(data), 1): NAs introduced by coercion
## Warning in FUN(newX[, i], ...): NAs introduced by coercion
## Warning in Lag(as.numeric(data), 1): NAs introduced by coercion
## Warning in FUN(newX[, i], ...): NAs introduced by coercion
## Warning in Lag(as.numeric(data), 1): NAs introduced by coercion
```

```
grdf <- cbind(gr$...98,gr$...99)
colnames(grdf)<-c("SF","SJ")

# seperate train set and test set
train_gr <- grdf[1:155, ]
test_gr <- grdf[156:176,]
# fit a var model
var_s=VAR(train_gr,p=2)
summary(var_s)
```

```
##
## VAR Estimation Results:
## ========================
## Endogenous variables: SF, SJ
## Deterministic variables: const
## Sample size: 153
## Log Likelihood: -258.804
## Roots of the characteristic polynomial:
## 0.8359 0.8348 0.8348 0.3391
## Call:
## VAR(y = train_gr, p = 2)
##
##
```

```
## Estimation results for equation SF:
## ===================================
## SF = SF.l1 + SJ.l1 + SF.l2 + SJ.l2 + const
##
##       Estimate Std. Error t value Pr(>|t|)
## SF.l1  0.93395    0.24868   3.756 0.000248 ***
## SJ.l1  0.23505    0.22706   1.035 0.302277
## SF.l2  0.02045    0.24881   0.082 0.934600
## SJ.l2 -0.31065    0.22644  -1.372 0.172170
## const  0.21659    0.10929   1.982 0.049369 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 1.117 on 148 degrees of freedom
## Multiple R-Squared: 0.8414,  Adjusted R-squared: 0.8371
## F-statistic: 196.3 on 4 and 148 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation SJ:
## ===================================
## SJ = SF.l1 + SJ.l1 + SF.l2 + SJ.l2 + const
##
##       Estimate Std. Error t value Pr(>|t|)
## SF.l1  -0.5669     0.2566  -2.209  0.02868 *
## SJ.l1   1.7298     0.2343   7.383 1.04e-11 ***
## SF.l2   0.7018     0.2567   2.733  0.00703 **
## SJ.l2  -1.0012     0.2336  -4.285 3.27e-05 ***
## const   0.2719     0.1128   2.411  0.01713 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 1.152 on 148 degrees of freedom
## Multiple R-Squared: 0.8365,  Adjusted R-squared: 0.8321
## F-statistic: 189.3 on 4 and 148 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##       SF    SJ
## SF 1.247 1.244
## SJ 1.244 1.327
##
## Correlation matrix of residuals:
##        SF     SJ
## SF 1.0000 0.9668
## SJ 0.9668 1.0000
```

```
AIC(var_s)
```

```
## [1] 537.6087
```

```r
BIC(var_s)
```

```
## [1] 567.9131
```

As the summary provided above, Var(2) is the best model to represent the relationship of the house price growth rates in this two cities, with AIC, BIC = 537.6087, 567.9131. For SF, only the lag 1 of SF has significance. For SJ, both two lags of SF and SJ have significant influences on current SJ housing price growth.

11.2

```r
# test if SJ affects SF. H0: SJ doesn't cause SF
grangertest(train_gr[,1] ~ train_gr[,2], order = 2)
```

```
## Granger causality test
##
## Model 1: train_gr[, 1] ~ Lags(train_gr[, 1], 1:2) + Lags(train_gr[, 2], 1:2)
## Model 2: train_gr[, 1] ~ Lags(train_gr[, 1], 1:2)
##   Res.Df Df      F Pr(>F)
## 1    148
## 2    150 -2 1.0257 0.3611
```

```r
# we fail to reject H0

# test if SF affects SJ. H0: SF doesn't affect SJ
grangertest(train_gr[,2] ~ train_gr[,1], order = 2)
```

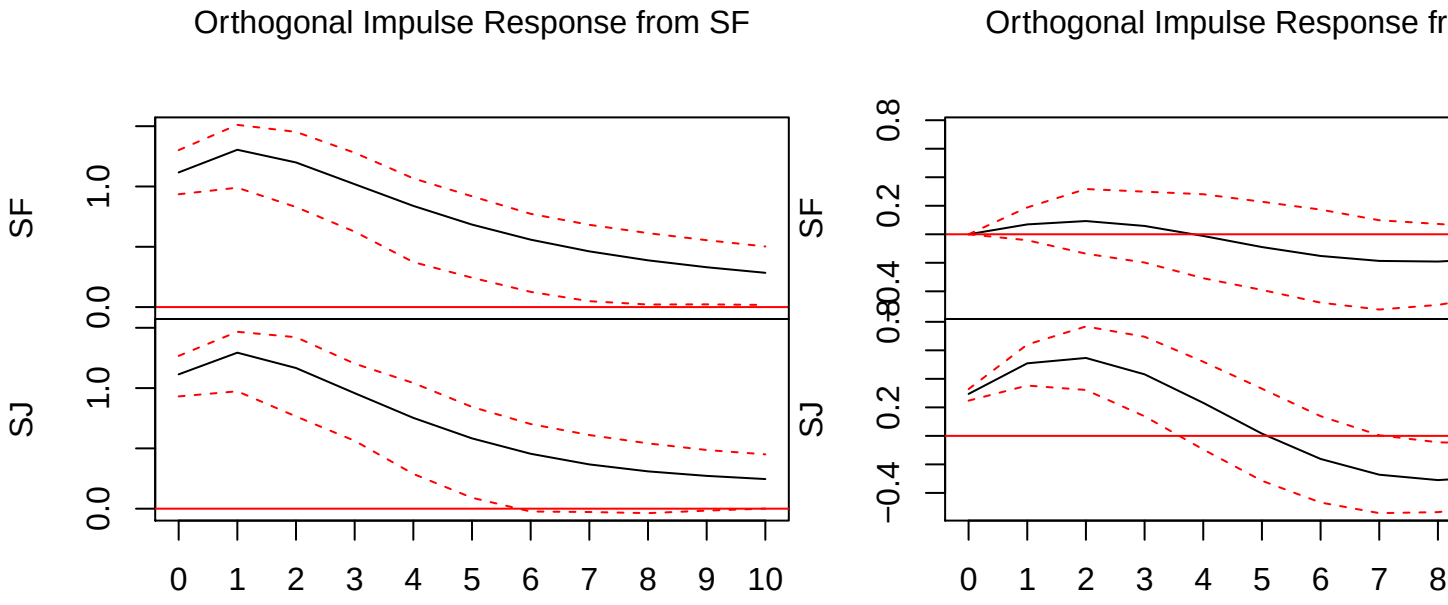
```
## Granger causality test
##
## Model 1: train_gr[, 2] ~ Lags(train_gr[, 2], 1:2) + Lags(train_gr[, 1], 1:2)
## Model 2: train_gr[, 2] ~ Lags(train_gr[, 2], 1:2)
##   Res.Df Df      F  Pr(>F)
## 1    148
## 2    150 -2 3.9378 0.02157 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# we can reject H0
```

From the granger causality test of order 2, we can see that the growth in SJ doesn't cause the growth in SF, but the growth in SF does cause the growth in SJ.

11.3

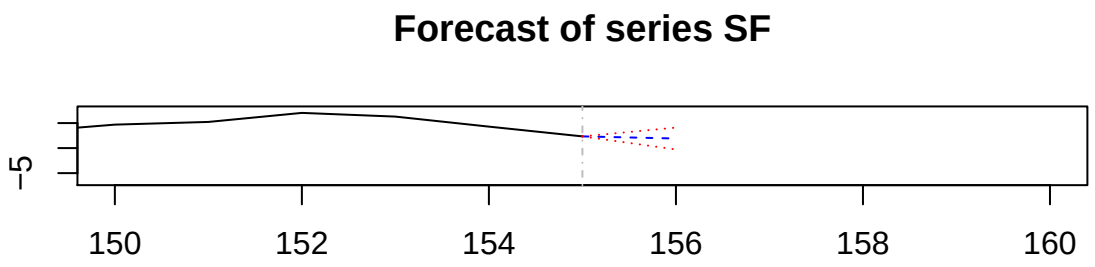
```r
# plot impulse response function
plot(irf(var_s))
```

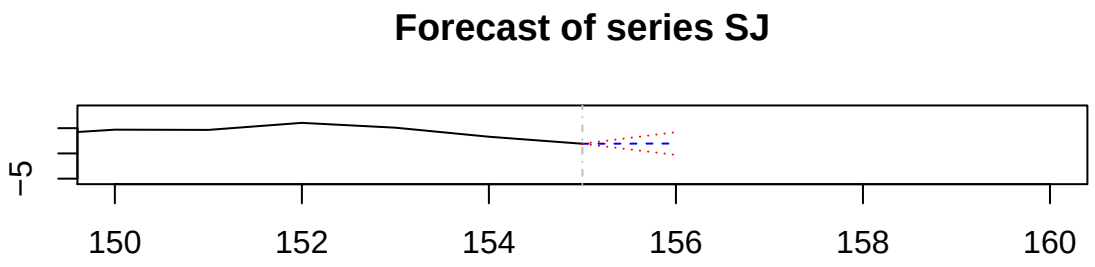


In first graph, the growth rate change in SF initially gives a large effect on both SJ and SF, and then decays slowly, and last over 10 quarters. In second graph, the growth rate change in SJ doesn't affect SF so much, and affects itself larger at first and then decays to negative. More job oppurtunies in SF but the housing is relatively high in SF. As a result, more people choose to live in SJ instead. The ordering matters because this is a one-way causality since SF dominant the economies.

11.4

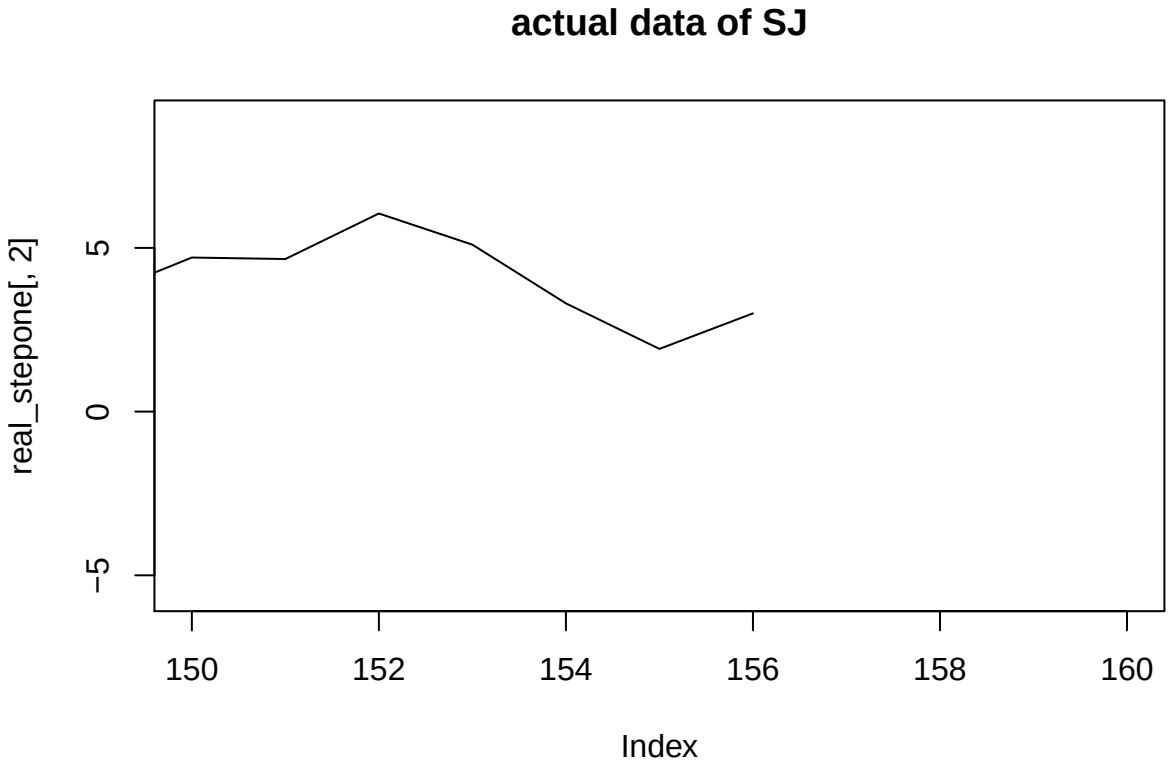
```
# forecast one step ahead
var_p<-predict(object=var_s, n.ahead=1)
plot(var_p,xlim=c(150,160))
```






```
# real data
real_stepone<-rbind(train_gr,test_gr[1,])
plot(real_stepone[,1],type='l',xlim=c(150,160),main="actual data of SF")
```


```
plot(real_stepone[,2],type='l',xlim=c(150,160),main="actual data of SJ")
```




```
# univariate model
## linear+lag
sf_m1<-dynlm(train_gr[,1]~Lag(train_gr[,1],1))
summary(sf_m1)
```

```
##
## Time series regression with "numeric" data:
## Start = 1, End = 154
##
## Call:
## dynlm(formula = train_gr[, 1] ~ Lag(train_gr[, 1], 1))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8922 -0.7319  0.0596  0.6528  4.2642
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            0.15515    0.11039   1.405    0.162
## Lag(train_gr[, 1], 1)  0.90711    0.03403  26.655   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.162 on 152 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.8238, Adjusted R-squared:  0.8226
## F-statistic: 710.5 on 1 and 152 DF,  p-value: < 2.2e-16
```

```
sj_m1<-dynlm(train_gr[,2]~Lag(train_gr[,2],1))
summary(sj_m1)
```

```
##
## Time series regression with "numeric" data:
## Start = 1, End = 154
##
## Call:
## dynlm(formula = train_gr[, 2] ~ Lag(train_gr[, 2], 1))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3540 -0.7697  0.1268  0.6373  4.1878
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            0.18973    0.12284   1.544    0.125
## Lag(train_gr[, 2], 1)  0.89122    0.03642  24.468   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.268 on 152 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.7975, Adjusted R-squared:  0.7962
## F-statistic: 598.7 on 1 and 152 DF,  p-value: < 2.2e-16
```

```
# use linear + lag to forecast one step ahead
sf_156<-sf_m1$coefficients[1]+sf_m1$coefficients[2]*train_gr[155,1]
sj_156<-sj_m1$coefficients[1]+sj_m1$coefficients[2]*train_gr[155,2]
print(sf_156)
```

```
## (Intercept)
##     2.27715
```

```
print(sj_156)
```

```
## (Intercept)
##    1.896525
```

```
## linear
ts_x<-seq(1975, 2013,length=length(train_gr[,1]))
m2<-lm(train_gr~ts_x)
summary(m2)
```

```
## Response SF :
##
## Call:
## lm(formula = SF ~ ts_x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.5907 -1.9270 -0.0749  1.9346  6.6192
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 144.27342   38.36057   3.761 0.000240 ***
## ts_x         -0.07149    0.01924  -3.716 0.000283 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 2.644 on 153 degrees of freedom
## Multiple R-squared:  0.08279,    Adjusted R-squared:  0.07679
## F-statistic: 13.81 on 1 and 153 DF,  p-value: 0.0002831
##
##
## Response SJ :
##
## Call:
## lm(formula = SJ ~ ts_x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.3784 -1.7179 -0.1077  1.6615  5.7571
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 153.53839   38.92730   3.944 0.000122 ***
## ts_x         -0.07606    0.01952  -3.896 0.000146 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.683 on 153 degrees of freedom
## Multiple R-squared:  0.09026,    Adjusted R-squared:  0.08431
## F-statistic: 15.18 on 1 and 153 DF,  p-value: 0.0001457
```

```
##  linear lag regression works better
AIC(sf_m1,sj_m1)
```

```
##       df      AIC
## sf_m1  3 487.2789
## sj_m1  3 514.0329
```

```
BIC(sf_m1,sj_m1)
```

```
##       df      BIC
## sf_m1  3 496.3897
## sj_m1  3 523.1438
```

```
# unconditional predictability test
# construct quadratic loss function
l1_sf<-MSE(sf_m1$fitted.values,train_gr[,1])
```

```
## Warning in y_true - y_pred: longer object length is not a multiple of
## shorter object length
```

```
l1_sj<-MSE(sj_m1$fitted.values,train_gr[,2])
```

```
## Warning in y_true - y_pred: longer object length is not a multiple of
## shorter object length
```

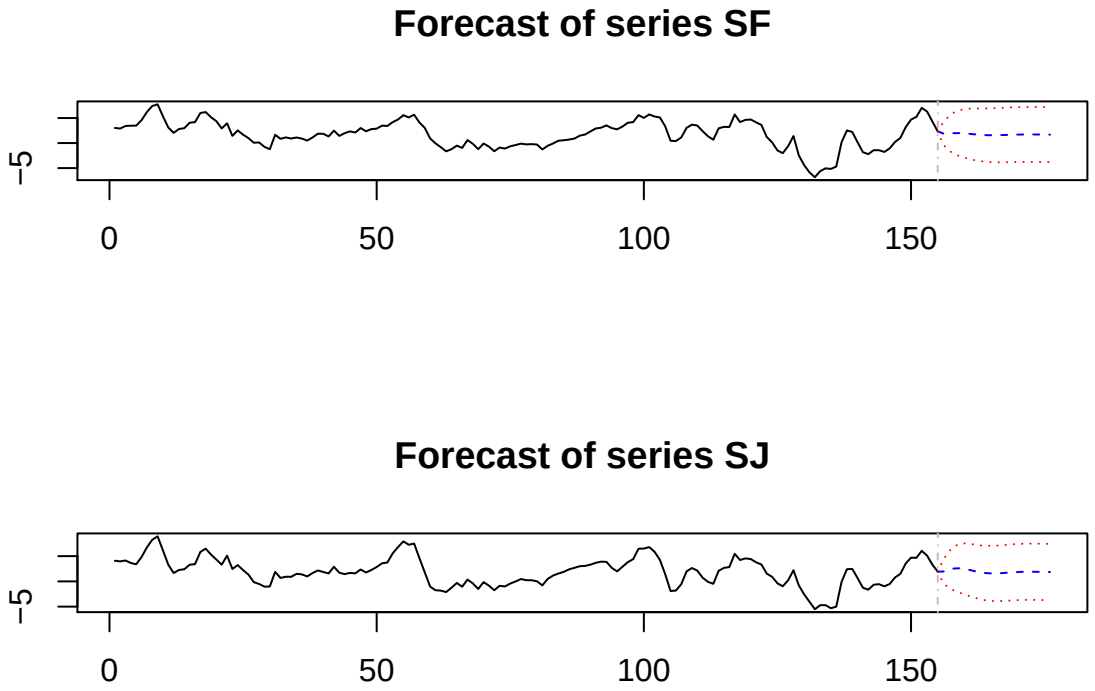
```
lv_sf<-MSE(var_p$endog[,1],train_gr[,1])
lv_sj<-MSE(var_p$endog[,2],train_gr[,2])

diff_sf<-l1_sf - lv_sf
diff_sj<-l1_sj - lv_sj
```

```
## both diff_sf and diff_sj are positive, which means lag and linear regression is better than var pred
```

###As the summary and AIC,BIC and the unconditional predictability test, we can see the linear regression with lag1 as regressor can provide a better approximation than prediction by var.

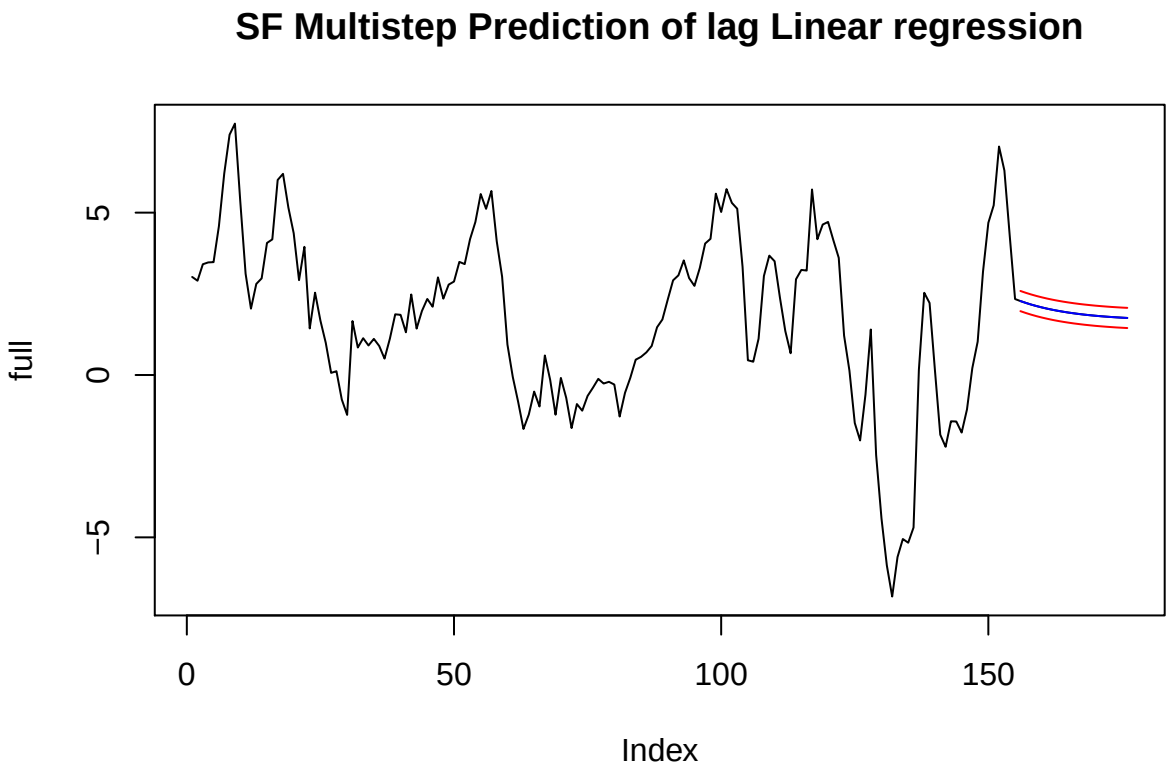
##11.5

```
# predict multiple steps with var
var_mp<-predict(object=var_s, n.ahead=21)
plot(var_mp)  # forecast with 95% interval
```

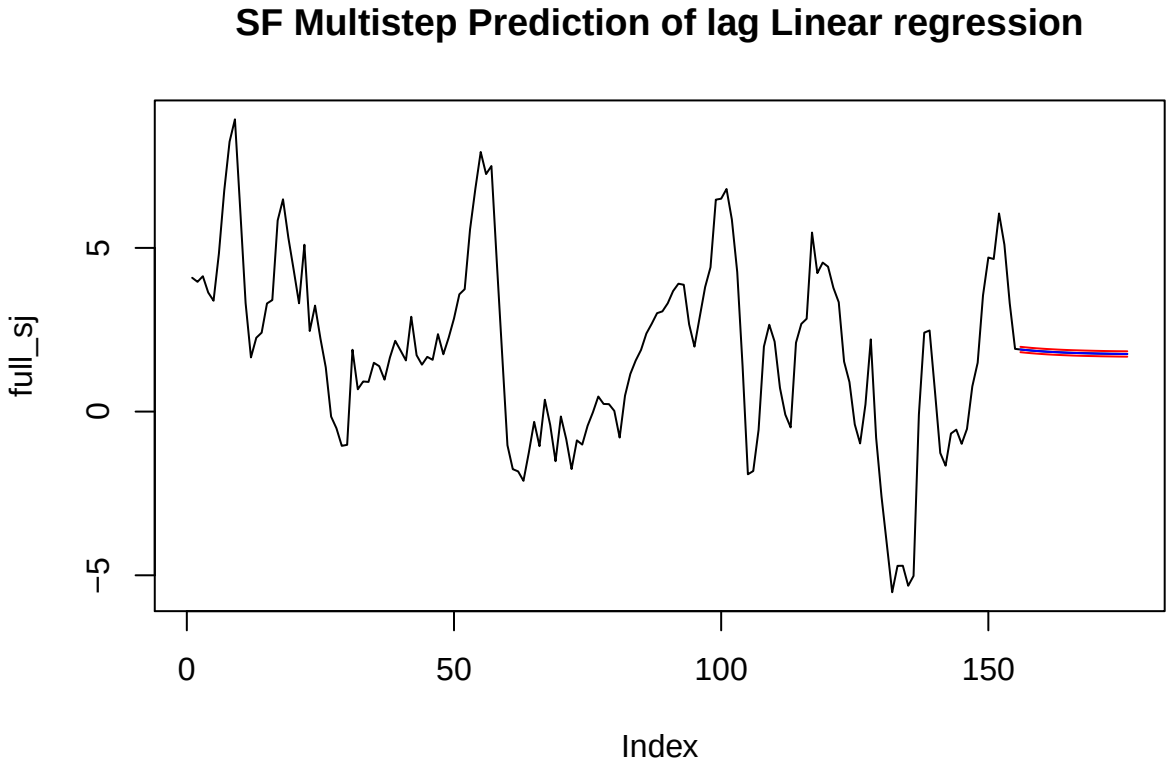



```
# predict multiple steps with linear regression with lag
sf_pred<-vector(mode="numeric",length=0)
sj_pred<-vector(mode="numeric",length=0)
sf_pred[1]<-sf_156
sj_pred[1]<-sj_156

for (i in 2:21)
{
  sf_pred[i]<-sf_m1$coefficients[1]+sf_m1$coefficients[2]*sf_pred[i-1]
  sj_pred[i]<-sj_m1$coefficients[1]+sj_m1$coefficients[2]*sj_pred[i-1]
}


sd_sf<-sd(sf_pred)
sd_sj<-sd(sj_pred)

p_x <- seq(156,176,by=1)

full <- c(train_gr[,1],sf_pred[])
full_sj <-c(train_gr[,2],sj_pred[])

plot(full,type='l',main="SF Multistep Prediction of lag Linear regression")
```

```
lines(x=p_x,y=sf_pred,col='blue')
lines(x=p_x,y=sf_pred+1.96*sd_sf,col="red")
lines(x=p_x,y=sf_pred-1.96*sd_sf,col="red")
```




```
plot(full_sj,type='l',main="SF Multistep Prediction of lag Linear regression")
lines(x=p_x,y=sj_pred,col='blue')
lines(x=p_x,y=sj_pred+1.96*sd_sj,col="red")
lines(x=p_x,y=sj_pred-1.96*sd_sj,col="red")
```




```
## data from albany-Sche
gras<-gr$...6 #growth rate
gras_train<-gras[1:155] # train set
gras_test<-gras[156:176] # test set

print(gras)
```

```
##   [1] -0.936298260  0.091318413 -1.415644158 -2.826254434 -2.174460655
##   [6] -0.903714119 -0.954352264  0.673412810  2.485061952  1.215137564
##  [11]  3.060084022  1.583963008  3.573779855  5.030950807  2.223905477
##  [16]  0.421819019  0.591309802  2.196414235  0.476242524  0.105189368
##  [21]  0.776694439  1.805663598  0.971547195  1.100254417 -0.207788242
##  [26]  2.433418562 -2.524376861 -0.912709151  1.947805691  0.324598803
##  [31]  1.880731199  2.096136574  1.472409374  3.520282410  3.727842506
##  [36]  1.802712608  4.093717376  2.357467973  1.950578129  3.201653750
##  [41]  2.368182230  3.597279539  4.430258511  6.367705207  5.689127460
##  [46]  5.234286403  6.083256761  4.173337208  4.512250849  3.528905255
##  [51]  3.025311721  2.332502011  2.876455090  1.661396088  0.862358986
##  [56]  1.791408355  0.899867049  0.901722377  0.703977682  0.367668540
##  [61]  0.141922450 -0.214026328 -0.353554083  0.015827576  0.142030258
##  [66]  0.111131702  0.310098083  1.384055603 -0.671968765  0.010423210
##  [71]  0.313255086 -1.679866670 -0.862966352 -0.832497088 -0.680895306
##  [76] -0.063892394  0.322750161 -0.847018403 -0.776935005 -0.533872966
##  [81] -1.441523388 -0.597722138 -0.499386265 -0.378549867 -0.166830603
##  [86] -0.151815422 -0.559138672  0.072677726  0.296753623 -0.761309986
##  [91]  0.005076451  0.041514729  0.929544644  0.084766059  0.645636061
##  [96]  0.866467837  0.312221388  1.290800865  0.366713814  0.757089486
## [101]  1.409189216  0.811236871  1.206907805  1.395152510  1.367265827
## [106]  1.893432795  1.665063420  2.025191210  2.154320436  3.020910549
## [111]  3.568382874  3.014410696  1.769101247  3.829681065  2.716866195
```

```
## [116]  3.178619088  3.837833726  3.128521801  2.963668058  3.083345663
## [121]  3.515787903  2.689730091  2.619946394  1.559523683  1.090211274
## [126]  0.069763592 -0.405766041  1.491029677 -0.394637670  0.019389238
## [131] -0.160077774 -0.313120426 -0.230804311 -0.523862830 -0.871584424
## [136] -1.625820356 -0.406021926  0.282984200  0.379842285  0.749128839
## [141] -0.721645984 -1.575503854  0.055392260 -0.462750109 -1.352689344
## [146] -0.208972145 -1.029240567  0.041202714  0.641949084  0.166957711
## [151]  0.705454708  0.417792012  0.566189222  0.676281641  0.082524850
## [156]  0.029814055 -0.385631165 -0.265791851  0.295930930  0.586021838
## [161]  0.381355604  0.461403579  0.400917806  0.560014648  0.849812615
## [166]  0.810422956  0.040661800  0.440593125  1.051548631  1.061174721
## [171]  1.709335754  1.150963454  0.612469869  0.596443364  0.224238354
## [176]  0.468094998
## attr(,"na.action")
## [1] 1 2
## attr(,"class")
## [1] "omit"
```

In the multistep prediction of Var, SF seems like a smooth horizontal line, while SJ has some fluctuations. In the multistep prediction of linear regression with lag, both SF and SJ are smooth and have a decreasing trend.

##11.6

```
## I choose to compare AS and SF
train_assf<-cbind(gras_train,train_gr[,1])
colnames(train_assf)<-c("AS","SF")
var_assf<-VAR(train_assf,p=2)
AIC(var_assf)
```

```
## [1] 899.7002
```

```
BIC(var_assf)
```

```
## [1] 930.0046
```

```
summary(var_assf)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: AS, SF
## Deterministic variables: const
## Sample size: 153
## Log Likelihood: -439.85
## Roots of the characteristic polynomial:
## 0.8951 0.7852 0.4039 0.3381
## Call:
## VAR(y = train_assf, p = 2)
##
##
## Estimation results for equation AS:
## ===================================
## AS = AS.l1 + SF.l1 + AS.l2 + SF.l2 + const
##
##       Estimate Std. Error t value Pr(>|t|)
```

```
## AS.l1  0.51016    0.08068   6.323 2.86e-09 ***
## SF.l1  0.13996    0.07327   1.910   0.0581 .
## AS.l2  0.33538    0.08039   4.172 5.13e-05 ***
## SF.l2 -0.10492    0.07389  -1.420   0.1577
## const  0.10478    0.10187   1.029   0.3054
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 1.001 on 148 degrees of freedom
## Multiple R-Squared: 0.6832,  Adjusted R-squared: 0.6746
## F-statistic: 79.79 on 4 and 148 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation SF:
## ===================================
## SF = AS.l1 + SF.l1 + AS.l2 + SF.l2 + const
##
##       Estimate Std. Error t value Pr(>|t|)
## AS.l1 -0.21150    0.08870  -2.384   0.0184 *
## SF.l1  1.23587    0.08056  15.342  < 2e-16 ***
## AS.l2  0.21052    0.08839   2.382   0.0185 *
## SF.l2 -0.35203    0.08124  -4.333  2.7e-05 ***
## const  0.19519    0.11200   1.743   0.0835 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 1.101 on 148 degrees of freedom
## Multiple R-Squared: 0.8458,  Adjusted R-squared: 0.8416
## F-statistic:   203 on 4 and 148 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##        AS     SF
## AS 1.0027 0.2541
## SF 0.2541 1.2120
##
## Correlation matrix of residuals:
##        AS     SF
## AS 1.0000 0.2305
## SF 0.2305 1.0000
```

```
# grangertest
as<-train_assf[,1]
sf<-train_assf[,2]

grangertest(as~sf, order = 2)
```

```
## Granger causality test
##
## Model 1: as ~ Lags(as, 1:2) + Lags(sf, 1:2)
## Model 2: as ~ Lags(as, 1:2)
##   Res.Df Df      F Pr(>F)
```

```
## 1    148
## 2    150 -2 2.0771 0.1289
```

```
grangertest(sf~as, order = 2)
```

```
## Granger causality test
##
## Model 1: sf ~ Lags(sf, 1:2) + Lags(as, 1:2)
## Model 2: sf ~ Lags(sf, 1:2)
##   Res.Df Df      F  Pr(>F)
## 1    148
## 2    150 -2 3.1768 0.04458 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

By attemping different values of p, I choose Var(2) which has the relatively smallest AIC,BIC. AS is only affected its own lag values, while SF is affected mostly by its own lag values, and slightly by AS's lag values. By taking Granger test, the growth in as has somehow affect the growth in sf.