Econ 211C HW 3

Anthony Fontana

June 2, 2017

Number 1

a

```
gam0<-17.5170
gam1<-15.9570
gam2<-12.4010
gam3<-8.3985
gam4<-5.0576
gam5<-3.0155

vec1<-c(gam0,gam1,gam2,gam3,gam4)
vec2<-c(gam1,gam0,gam1,gam2,gam3)
vec3<-c(gam2,gam1,gam0,gam1,gam2)
vec4<-c(gam3,gam2,gam1,gam0,gam1)
vec5<-c(gam4,gam3,gam2,gam1,gam0)

colvec<-t(c(gam1,gam2,gam3,gam4,gam5))

mat1<-rbind(vec1,vec2,vec3,vec4,vec5)

betas<-solve(mat1)%*%t(colvec)
```

b/c

```
#b

func1<-function(n){

phi<-c(1.3,-0.4)
theta<-c(0.7,0,0.1,-0.5,-0.2)
p<-length(phi)
q<-length(theta)
r<-max(p,q)
err<-rnorm(n+r,0,1)
y<-rep(0,n+r)
for( i in (r+1):(r+n)){
  y[i]<-err[i]+t(phi)%*%y[(i-1):(i-p)]+t(theta)%*%err[(i-1):(i-q)]
}
y<-y[-(1:r)]
}



#C

#create new vector containing forecasts

y<-func1(105)
forecastvec<-rep(0,5)
forecastvec[1]<-betas[1]*y[100]+betas[2]*y[99]+betas[3]*y[98]+betas[4]*y[97]+betas[5]*y[96]
forecastvec[2]<-
betas[1]*forecastvec[1]+betas[2]*y[100]+betas[3]*y[99]+betas[4]*y[98]+betas[5]*y[97]
forecastvec[3]<-betas[1]*forecastvec[2]+betas[2]*forecastvec[1]+betas[3]*y[100]+betas[4]*y[99]+b
etas[5]*y[98]
forecastvec[4]<-
betas[1]*forecastvec[3]+betas[2]*forecastvec[2]+betas[3]*forecastvec[1]+betas[4]*y[100]+betas[5]*y
9]
forecastvec[5]<-
betas[1]*forecastvec[4]+betas[2]*forecastvec[3]+betas[3]*forecastvec[2]+betas[4]*forecastvec[1]+b
etas[5]*y[100]

forecastvec
```

```
## [1] -5.1621641 -3.4707394 -1.9364587 -0.9122664 -0.5240398
```

```
meansqerr<-rep(0,1000)
for(i in 1:1000){
  y<-func1(105)
  forecastvec<-rep(0,5)
  forecastvec[1]<-betas[1]*y[100]+betas[2]*y[99]+betas[3]*y[98]+betas[4]*y[97]+betas[5]*y[96]
  forecastvec[2]<-
betas[1]*forecastvec[1]+betas[2]*y[100]+betas[3]*y[99]+betas[4]*y[98]+betas[5]*y[97]
  forecastvec[3]<-
betas[1]*forecastvec[2]+betas[2]*forecastvec[1]+betas[3]*y[100]+betas[4]*y[99]+betas[5]*y[98]
  forecastvec[4]<-
betas[1]*forecastvec[3]+betas[2]*forecastvec[2]+betas[3]*forecastvec[1]+betas[4]*y[100]+betas[5]*y
9]
  forecastvec[5]<-
betas[1]*forecastvec[4]+betas[2]*forecastvec[3]+betas[3]*forecastvec[2]+betas[4]*forecastvec[1]+b
etas[5]*y[100]
  meansqerr[i]<-(forecastvec[5]-y[105])^2

}

mse<-mean(meansqerr)
mse
```

```
## [1] 17.55869
```

d/e

```
#d


autocov<-c(gam0,gam1,gam2,gam3,gam4,gam5,0,0,0,0)
for(i in 7:10){
  autocov[i]<-1.3*autocov[i-1]-0.4*autocov[i-2]
}
vector1<-as.matrix(autocov[6:10],nrow=5)
betas1<-solve(mat1)%*%vector1
betas1
```

```
##             [,1]
## [1,]  0.66058023
## [2,] -0.78791078
## [3,]  0.32258357
## [4,]  0.01081816
## [5,] -0.01471187
```

```
#e

##Five Step Forecast

forecastvec2<-betas1[1]*y[100]+betas1[2]*y[99]+betas1[3]*y[98]+betas1[4]*y[97]+betas1[5]*y[96]
forecastvec2
```

```
## [1] 0.4438863
```

f

```
#f

### 5 Step Squared Error

meansqerr5<-rep(0,1000)
for(i in 1:1000){
  y<-func1(105)
  forecastvec2<-betas1[1]*y[100]+betas1[2]*y[99]+betas1[3]*y[98]+betas1[4]*y[97]+betas1[5]*y[96]
  meansqerr5<-(forecastvec2-y[105])^2
}

mean(meansqerr5)
```

```
## [1] 1.706941
```

g

```
#g

#### Same procedure but using coef from arima


library(forecast)

y<-func1(105)
reg1<-arima(y[1:100],order=c(2,0,5))
reg1
```

```
##
## Call:
## arima(x = y[1:100], order = c(2, 0, 5))
##
## Coefficients:
##          ar1      ar2     ma1     ma2     ma3     ma4     ma5  intercept
##       0.7971  -0.4997  0.9656  1.0880  1.2443  0.6587  0.5367     0.2963
## s.e.  0.1554   0.1477  0.1375  0.1997  0.1511  0.1633  0.1537     0.6875
##
## sigma^2 estimated as 0.8025:  log likelihood = -134.74,  aic = 287.49
```

h

```
### obtain the coefficients from the regression



phi<-c(reg1$coef[1],reg1$coef[2])
theta<-c(reg1$coef[3],reg1$coef[4],reg1$coef[5],reg1$coef[6],reg1$coef[7])

#Autcorrelations
autos<-ARMAacf(ar=phi,ma=theta,lag.max=10)
#Autocovs
autocovs<-autos*var(y)

col1<-c(autocovs[1],autocovs[2],autocovs[3],autocovs[4],autocovs[5])
col2<-c(autocovs[2],autocovs[1],autocovs[2],autocovs[3],autocovs[4])
col3<-c(autocovs[3],autocovs[2],autocovs[1],autocovs[2],autocovs[3])
col4<-c(autocovs[4],autocovs[3],autocovs[2],autocovs[1],autocovs[2])
col5<-c(autocovs[5],autocovs[4],autocovs[3],autocovs[2],autocovs[1])

mat2<-rbind(col1,col2,col3,col4,col5)
mat2
```

```
##               0         1         2         3          4
## col1 11.9336823 10.508048  7.238084  3.490709  0.5379844
## col2 10.5080480 11.933682 10.508048  7.238084  3.4907090
## col3  7.2380840 10.508048 11.933682 10.508048  7.2380840
## col4  3.4907090  7.238084 10.508048 11.933682 10.5080480
## col5  0.5379844  3.490709  7.238084 10.508048 11.9336823
```

```
gams<-matrix(c(autocovs[2],autocovs[3],autocovs[4],autocovs[5],autocovs[6]),nrow=5)
gams
```

```
##            [,1]
## [1,] 10.5080480
## [2,]  7.2380840
## [3,]  3.4907090
## [4,]  0.5379844
## [5,] -0.8564702
```

```
betas2<-solve(mat2)%*%gams
betas2
```

```
##          [,1]
## 0  1.62375748
## 1 -0.82937963
## 2 -0.01535625
## 3 -0.03333840
## 4  0.13630058
```

I/J

```
###New forecasts
y<-func1(105)
newforecast<-rep(0,5)
for(i in 1:5){
  newforecast[i]<-
betas2[1]*y[99+i]+betas2[2]*y[98+i]+betas2[3]*y[97+i]+betas2[4]*y[96+i]+betas2[5]*y[95+i]
}

newsqerr<-rep(0,1000)
for(j in 1:1000){
  y<-func1(105)
  for(i in 1:5){
    newforecast[i]<-betas2[1]*y[99+i]+betas2[2]*y[98+i]+betas2[3]*y[97+i]+betas2[4]*y[96+i]+beta
s2[5]*y[95+i]
  }
  newsqerr[j]<-(newforecast[5]-y[105])^2
}
mean(newsqerr)
```

```
## [1] 1.338047
```

```
###
y<-func1(105)
forecast_<-rep(0,5)
forecast_[1]<-betas2[1]*y[100]+betas2[2]*y[99]+betas2[3]*y[98]+betas2[4]*y[97]+betas2[5]*y[96]
forecast_[2]<-
betas2[1]*forecast_[1]+betas2[2]*y[100]+betas2[3]*y[99]+betas2[4]*y[98]+betas2[5]*y[97]
forecast_[3]<-betas2[1]*forecast_[2]+betas2[2]*forecast_[1]+betas2[3]*y[100]+betas[4]*y[99]+beta
s[5]*y[98]
forecast_[4]<-betas2[1]*forecast_[3]+betas2[2]*forecast_[2]+betas2[3]*forecast_[1]+betas2[4]*y[1
00]+betas2[5]*y[99]
forecast_[5]<-betas2[1]*forecast_[4]+betas2[2]*forecast_[3]+betas2[3]*forecast_[2]+betas2[4]*for
ecast_[1]+betas2[5]*y[100]

forecast_
```

```
## [1] -1.35400592 -0.87431903 -0.06245055  0.43533973  0.56675185
```

```
anothermeansqerr<-rep(0,1000)
for(i in 1:1000){
  y<-func1(105)
  forecast_<-rep(0,5)
  forecast_[1]<-betas2[1]*y[100]+betas2[2]*y[99]+betas2[3]*y[98]+betas2[4]*y[97]+betas2[5]*y[96]
  forecast_[2]<-
betas2[1]*forecast_[1]+betas2[2]*y[100]+betas2[3]*y[99]+betas2[4]*y[98]+betas2[5]*y[97]
  forecast_[3]<-betas2[1]*forecast_[2]+betas2[2]*forecast_[1]+betas2[3]*y[100]+betas[4]*y[99]+be
tas[5]*y[98]
  forecast_[4]<-
betas2[1]*forecast_[3]+betas2[2]*forecast_[2]+betas2[3]*forecast_[1]+betas2[4]*y[100]+betas2[5]*y[
9]
  forecast_[5]<-betas2[1]*forecast_[4]+betas2[2]*forecast_[3]+betas2[3]*forecast_[2]+betas2[4]*f
orecast_[1]+betas2[5]*y[100]
  anothermeansqerr[i]<-(forecast_[5]-y[105])^2

}

mean(anothermeansqerr)
```

```
## [1] 22.91601
```

K/L

```
### Part D again
vec_<-matrix(NA,nrow=5,ncol=1)
for(i in 1:5){
  vec_[i,1]<-autocovs[i+5]
}

betas3<-solve(mat2)%*%vec_
betas3
```

```
##         [,1]
## 0  0.3668678
## 1 -0.8153995
## 2  0.3169569
## 3  0.3342672
## 4 -0.2365998
```

```
anotherforecast<-
betas3[1]*y[100]+betas3[2]*y[99]+betas3[3]*y[98]+betas3[4]*y[97]+betas3[5]*y[96]
anotherforecast
```

```
## [1] -1.310898
```

```
squarerror<-rep(NA,1000)
for(i in 1:1000){
  y<-func1(105)
  forecastvec_=betas3[1]*y[100]+betas3[2]*y[99]+betas3[3]*y[98]+betas3[4]*y[97]+betas3[5]*y[96]
  squarerror[i]<-(forecast_-y[105])^2
}
mean(squarerror)
```

```
## [1] 52.31359
```

Number 2

a

```
#a

d<-read.csv("C:/Users/Aj/Documents/UCSC Coursework/Spring Quarter 2017/Econ 211C - Time Series/H
omework/Assignment3Data.csv")
size<-nrow(d)
reg1<-lm(d$Returns[3:size]~d$Returns[2:(size-1)]+d$Returns[1:(size-2)]+d$OrderFlow[2:(size-
1)]+d$OrderFlow[1:(size-2)])
reg2<-lm(d$OrderFlow[3:size]~d$OrderFlow[2:(size-1)]+d$OrderFlow[1:(size-2)]+d$Returns[2:(size-
1)]+d$Returns[1:(size-2)])

summary(reg1)
```

```
##
## Call:
## lm(formula = d$Returns[3:size] ~ d$Returns[2:(size - 1)] + d$Returns[1:(size -
##     2)] + d$OrderFlow[2:(size - 1)] + d$OrderFlow[1:(size - 2)])
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -7.940e-04 -1.120e-04 -8.060e-06  9.688e-05  1.977e-03
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 8.057e-06  1.213e-05   0.664    0.507
## d$Returns[2:(size - 1)]    -5.967e-03  7.470e-02  -0.080    0.936
## d$Returns[1:(size - 2)]     6.708e-02  7.440e-02   0.902    0.368
## d$OrderFlow[2:(size - 1)] -1.195e-07  1.327e-07  -0.900    0.369
## d$OrderFlow[1:(size - 2)] -4.342e-08  1.314e-07  -0.331    0.741
##
## Residual standard error: 0.0002386 on 383 degrees of freedom
## Multiple R-squared:  0.007863,   Adjusted R-squared:  -0.002499
## F-statistic: 0.7588 on 4 and 383 DF,  p-value: 0.5526
```

```
summary(reg2)
```

```
##
## Call:
## lm(formula = d$OrderFlow[3:size] ~ d$OrderFlow[2:(size - 1)] +
##     d$OrderFlow[1:(size - 2)] + d$Returns[2:(size - 1)] + d$Returns[1:(size -
##     2)])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -599.50  -31.37   -3.97   30.41 1403.09
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 4.902e+00  6.804e+00   0.720    0.472
## d$OrderFlow[2:(size - 1)] -1.079e-01  7.448e-02  -1.449    0.148
## d$OrderFlow[1:(size - 2)]  5.419e-02  7.370e-02   0.735    0.463
## d$Returns[2:(size - 1)]    3.619e+04  4.191e+04   0.863    0.388
## d$Returns[1:(size - 2)]    2.830e+04  4.175e+04   0.678    0.498
##
## Residual standard error: 133.9 on 383 degrees of freedom
## Multiple R-squared:  0.01527,    Adjusted R-squared:  0.004983
## F-statistic: 1.484 on 4 and 383 DF,  p-value: 0.2062
```

C

```
#c
error<-matrix(0,nrow=2,ncol=length(residuals(reg2)))
error[1,]<-residuals(reg1)
error[2,]<-residuals(reg2)



#Variance and Covariance matrix of the residuals and find the eigenvectors
covmat<-matrix(NA,nrow=2,ncol=2)
covmat[1,1]=var(residuals(reg1))
covmat[2,2]=var(residuals(reg2))
covmat[1,2]=cov(residuals(reg1),residuals(reg2))
covmat[2,1]=cov(residuals(reg1),residuals(reg2))

covmat
```

```
##              [,1]         [,2]
## [1,] 5.634283e-08 2.311864e-02
## [2,] 2.311864e-02 1.773753e+04
```

```
eigenvec<-eigen(covmat)
vec_1<-eigenvec$vectors[,1]
vec_2<-eigenvec$vectors[,2]

mat_<-rbind(vec_1,vec_2)

newvec<-t(mat_)%*%error

## Mat_ is the matrix that orthogonalizes the error vector. This gets rid of contemporaneous cor
relation.
```