

Macroeconometria - Séries de tempo

FAUSTO JOSÉ ARAÚJO VIEIRA

Aula 9

17 de abril a 22 de maio de 2018

Aplicação dos modelos de série de tempo

Atualizando os dados

- ▶ `setwd("F:/series de tempo/aula_pratica")`
- ▶ `list.files()`
- ▶ `X<-read.csv("aula_aplic.csv",sep=";", dec=".", head=TRUE)`
- ▶ `X$date<-as.Date(X$date,'%d/%b/%y')`
- ▶ `head(X)`

Raízes unitárias

- ▶ `require(fUnitRoots)`
- ▶ `plot(X$arrec_rfb,type="l")`
- ▶ `adfTest(X$arrec_rfb,lags=2,type = "ct")`
- ▶ `adfTest(X$arrec_rfb,lags=2,type = "c")`

Filtro HP ou diferença

- ▶ `require(mFilter)`
- ▶ `pib_1600=hpfilter(X$pib_sa,type="lambda",freq = 1600)`
- ▶ `pib_14400=hpfilter(X$pib_sa,type="lambda",freq = 14400)`

Ajuste sazonal - Sarima

- ▶ `require(astsa)`
- ▶ `acf2((X$arrec_rfb))`
- ▶ `acf2(diff(X$arrec_rfb))`

Estimar o modelo

- ▶ `arrecad_sa<-sarima(X$arrec_rfb,0,1,1,0,1,1,4)`
- ▶ `arrecad_sa<-sarima(X$arrec_rfb,0,1,1,0,1,2,4)`

Projetar

- ▶ `arrecad_sarima<-sarima.for(X$arrec_rfb,8,0,1,1,0,1,2,4)`
- ▶ `arrecad_sarimaf<- arrecad_sarima$pred`

ARIMA com modelo ajustado sazonalmente

- ▶ `require(seasonal)`

Ajuste sazonal

- ▶ `arrecad<-ts(X$arrec_rfb,start=c(2004,1),frequency=4)`

- ▶ `arrecad_x11=seas(x=arrecad,transform.function = "log",x11="")`

- ▶ `summary(arrecad_x11)`

- ▶ `teste<-udg(arrecad_x11)`

- ▶ Kruskal-Wallis - `teste$f2.kw`

- ▶ `teste$f3.q`

- ▶ `arrecad_x11sa=arrecad_x11$series$d11`

ARIMA com modelo ajustado sazonalmente

Raiz unitária

- ▶ `adfTest(arrecad_x11sa,lags=2,type = "ct")`
- ▶ `adfTest(arrecad_x11sa,lags=2,type = "c")`

ACF e PACF

- ▶ `acf2(diff(arrecad_x11sa))`
- ▶ `require(FitARMA)`
- ▶ `regarma<- FitARMA(diff(arrecad_x11sa),order=c(0,0,1))`
- ▶ `coef(regarma)`
- ▶ `acf2(resid(regarma))` ou `regarma$LjungBoxQ`

ARIMA com modelo ajustado sazonalmente

Pacote para projetar

- ▶ `install.packages("forecast")`
- ▶ `require(forecast)`
- ▶ `regarma1<-Arima(arrecad_x11sa,c(0,0,1))`
- ▶ `arrecad_x11saf<-forecast(regarma)`
- ▶ `plot(arrecad_x11saf)`

Equação degenerada

- ▶ `install.packages("dyn")`
- ▶ `require(dyn)`
- ▶ `x1<-dyn$lm(diff(arrecad_x11sa)~lag(diff(arrecad_x11sa),-1))`
- ▶ `pib<-ts(X$pib_sa,start=c(2004,1),frequency=4)`
 - ▶ `ccf(diff(arrecad_x11sa),pib_gap)`
- ▶ `pib_gap=ts(pib_1600$cycle,start = c(2004,1),frequency=4)`
 - ▶ `ccf(diff(arrecad_x11sa),pib_gap)`
- ▶ `pim<-ts(X$pim_sa,start = c(2004,1),frequency=4)`
 - ▶ `ccf(diff(arrecad_x11sa),diff(pim))`
- ▶ `cagedsa<-ts(X$CAGED_sa,start = c(2004,1),frequency=4)`
 - ▶ `ccf(diff(arrecad_x11sa),diff(cagedsa))`

Equação degenerada

Estimando a equação

- ▶ `x1a<-dyn$lm(diff(arrecad_x11sa)~lag(diff(arrecad_x11sa),-1)+lag(diff(pim),-1)+lag(diff(cagedsa),0))`
- ▶ `summary(x1a)`
- ▶ `acf2(x1a$residuals)`

Equação degenerada

Criar a base com as hipóteses de crescimento

- ▶ `cagedsa_for=seq(from = 80000, to = 100000, length.out = 8)`
- ▶ `cagedsa_for=ts(cagedsa_for,start = c(2018,1),frequency = 4)`
- ▶ `pim_for=seq(from = 0.8, to = 1.5, length.out = 8)`
- ▶ `pim_for=ts(pim_for,start = c(2018,1),frequency = 4)`
- ▶ `dados_for<-cbind(pim_for,cagedsa_for)`

Projetando com as variáveis adicionais

- ▶ `arrecad_degf=predict(x1a,dados_for,n.ahead=8)`

VAR

- ▶ `require(vars)`
- ▶ `var_arrecad=cbind(pib_gap,diff(pim),diff(cagedsa),diff(arrecad_x11sa))`
- ▶ `colnames(var_arrecad)=c("hiato","ind","emp","arrec")`

```
var_arrecad[1,2]=1.00
```

```
var_arrecad[1,3]=10000
```

```
var_arrecad[1,4]=0
```

VAR

- ▶ `var_est<-VAR(var_arrecad,lag.max = 4,ic="SC")`
- ▶ `summary(var_est)`
- ▶ `var_est$p`
- ▶ `roots(var_est)`
- ▶ `serial.test(var_est,lags.pt=2,type = "PT.adjusted")`
- ▶ `serial.test(var_est,lags.pt=5,type = "PT.adjusted")`

VAR

Função resposta ao impulso

- ▶ `a=irf(var_est, impulse = c("emp", "ind"), response = "arrec", boot = TRUE,ci=0.95)`
- ▶ `plot(a)`

Projeção

- ▶ `var_proj<-predict(var_est,n.ahead=8,ci=0.95)`
- ▶ `arread_varf=var_proj$fcst$arrec`

VECM

- ▶ `require(tsDyn)`
- ▶ `dados=cbind(log(pib),log(pim),cagedsa,log(arrecad_x11sa))`
- ▶ `colnames(dados)=c("pib","pim","caged","arrecad")`
- ▶ `rank.select(dados,lag.max = 10)`
- ▶ `mod_vecm <- VECM(dados, lag=1, estim="ML")`
- ▶ `mod_vecm$model.specific`
- ▶ `rank.test(mod_vecm,type = "trace",r_null=0)`
- ▶ `rank.test(mod_vecm,type = "trace",r_null=1)`
- ▶ `rank.test(mod_vecm,type = "trace",r_null=2)`

VECM

Função resposta ao impulso

- ▶ `irf_g=irf(mod_vecm,impulse=c("pim","caged"),response="arrecad",boot=TRUE,ci=0.95)`
- ▶ `plot(irf_g)`

Projetando usando o VECM

- ▶ `vecm_proj<- predict(mod_vecm, n.ahead = 10)`
- ▶ `arrecad_vecmf=exp(vecm_proj[,4])`

Sistema

- ▶ `require("systemfit")`
- ▶ `X$arrecad_sa=arrecad_x11sa`

Sistema

- ▶ `eqProd<-pib_1600$cycle[-1]~pib_1600$cycle[-length(pib_1600$cycle)]+X$CAGED_sa[-1]+X$juros_cons[-length(X$juros_cons)]`
- ▶ `eqEmpr<-X$CAGED_sa[-1]~pib_1600$cycle[-1]+X$sal_adm_caged[-1]`
- ▶ `eqArrec<-X$arrecad_sa[-1]~X$arrecad_sa[-length(X$arrecad_sa)]+pib_1600$cycle[-length(pib_1600$cycle)]+X$CAGED_sa[-1]`

Sistema

- ▶ `system<-list(oferta=eqProd,empr=eqEmpr,arrecad=eqArrec)`
- ▶ `ajustsur<-systemfit(system,"SUR",data=X)`
- ▶ `ci<-confint(ajustsur)`
- ▶ `print(ci,digits=4)`