Tugas Praktikum Pertemuan 3 - Regresi dengan Peubah Lag

Adinda Shabrina Putri Salsabila (G1401221081)

## *Packages*

library(dLagM)  
library(dynlm)  
library(MLmetrics)  
library(lmtest)  
library(car)

# Data

Data diambil dari ouworldindata dengan peubah x merupakan data populasi Indonesia tahun 1962-2022 dan peubah y merupakan data kadar emisi karbon di Indonesia tahun 1962-2022.

data <- rio::import("https://raw.githubusercontent.com/adindashabrina/dataMPDW/main/Data-Populasi-dan-Emisi-Karbon.csv")  
str(data)

## 'data.frame': 60 obs. of 4 variables:  
## $ t : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ Yt : num 0.246 0.237 0.227 0.244 0.226 ...  
## $ Y(t-1): num NA 0.246 0.237 0.227 0.244 ...  
## $ Xt : num 0.0934 0.0961 0.0988 0.1014 0.1038 ...

data

## t Yt Y(t-1) Xt  
## 1 1 0.2461275 NA 0.09337585  
## 2 2 0.2372082 0.2461275 0.09605142  
## 3 3 0.2267469 0.2372082 0.09883376  
## 4 4 0.2438587 0.2267469 0.10136513  
## 5 5 0.2257117 0.2438587 0.10379276  
## 6 6 0.2309418 0.2257117 0.10652640  
## 7 7 0.2525465 0.2309418 0.10945001  
## 8 8 0.2974805 0.2525465 0.11251764  
## 9 9 0.3106125 0.2974805 0.11565750  
## 10 10 0.3291610 0.3106125 0.11883370  
## 11 11 0.3564182 0.3291610 0.12203984  
## 12 12 0.3937011 0.3564182 0.12528852  
## 13 13 0.4003785 0.3937011 0.12855505  
## 14 14 0.4110298 0.4003785 0.13184385  
## 15 15 0.4593204 0.4110298 0.13517366  
## 16 16 0.5978202 0.4593204 0.13853354  
## 17 17 0.6650441 0.5978202 0.14195316  
## 18 18 0.6575090 0.6650441 0.14543483  
## 19 19 0.6402667 0.6575090 0.14895054  
## 20 20 0.6612135 0.6402667 0.15248504  
## 21 21 0.6801835 0.6612135 0.15605215  
## 22 22 0.6624060 0.6801835 0.15965138  
## 23 23 0.6927793 0.6624060 0.16325112  
## 24 24 0.7334347 0.6927793 0.16677618  
## 25 25 0.7222980 0.7334347 0.17017506  
## 26 26 0.7184635 0.7222980 0.17351116  
## 27 27 0.7554084 0.7184635 0.17685507  
## 28 28 0.7358152 0.7554084 0.18020164  
## 29 29 0.8513446 0.7358152 0.18350110  
## 30 30 0.9433742 0.8513446 0.18677824  
## 31 31 1.0575962 0.9433742 0.19004374  
## 32 32 1.1218932 1.0575962 0.19330517  
## 33 33 1.1253953 1.1218932 0.19659183  
## 34 34 1.1224906 1.1253953 0.19988805  
## 35 35 1.2566929 1.1224906 0.20320435  
## 36 36 1.3680507 1.2566929 0.20653609  
## 37 37 1.1764085 1.3680507 0.20982679  
## 38 38 1.3839109 1.1764085 0.21300467  
## 39 39 1.3141832 1.3839109 0.21607779  
## 40 40 1.4601983 1.3141832 0.21909791  
## 41 41 1.4014755 1.4601983 0.22208850  
## 42 42 1.5212358 1.4014755 0.22504800  
## 43 43 1.5182068 1.5212358 0.22792665  
## 44 44 1.5192718 1.5182068 0.23087165  
## 45 45 1.4954154 1.5192718 0.23395165  
## 46 46 1.6514556 1.4954154 0.23706234  
## 47 47 1.5370413 1.6514556 0.24015790  
## 48 48 1.6554897 1.5370413 0.24322002  
## 49 49 1.8269529 1.6554897 0.24630533  
## 50 50 2.0264078 1.8269529 0.24947003  
## 51 51 2.0619888 2.0264078 0.25269853  
## 52 52 1.9309183 2.0619888 0.25585247  
## 53 53 1.9041111 1.9309183 0.25887740  
## 54 54 2.0809183 1.9041111 0.26179925  
## 55 55 2.0625758 2.0809183 0.26462743  
## 56 56 2.1056583 2.0625758 0.26734665  
## 57 57 2.2245428 2.1056583 0.26995185  
## 58 58 2.4144928 2.2245428 0.27248938  
## 59 59 2.2290483 2.4144928 0.27481486  
## 60 60 2.2499223 2.2290483 0.27675806

## Pembagian Data

#SPLIT DATA  
train<-data[1:40,]  
test<-data[41:60,]

#data time series  
train.ts<-ts(train)  
test.ts<-ts(test)  
data.ts<-ts(data)

# Model Koyck

Model Koyck didasarkan pada asumsi bahwa semakin jauh jarak lag peubah independen dari periode sekarang maka semakin kecil pengaruh peubah lag terhadap peubah dependen.

Koyck mengusulkan suatu metode untuk menduga model dinamis distributed lag dengan mengasumsikan bahwa semua koefisien mempunyai tanda sama.

Model kyock merupakan jenis paling umum dari model infinite distributed lag dan juga dikenal sebagai geometric lag

dengan

### Pemodelan

Pemodelan model Koyck dengan R dapat menggunakan dLagM::koyckDlm() . Fungsi umum dari koyckDlm adalah sebagai berikut.

koyckDlm(x , y , intercept)

Fungsi koyckDlm() akan menerapkan model lag terdistribusi dengan transformasi Koyck satu prediktor. Nilai x dan y tidak perlu sebagai objek *time series* (ts). intercept dapat dibuat TRUE untuk memasukkan intersep ke dalam model.

#MODEL KOYCK  
model.koyck <- koyckDlm(x = train$Xt, y = train$Yt)  
summary(model.koyck)

##   
## Call:  
## "Y ~ (Intercept) + Y.1 + X.t"  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.181364 -0.026090 -0.002304 0.025187 0.147600   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.2711 0.1032 -2.628 0.01255 \*   
## Y.1 0.6881 0.1208 5.695 1.78e-06 \*\*\*  
## X.t 3.2764 1.1515 2.845 0.00728 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.0628 on 36 degrees of freedom  
## Multiple R-Squared: 0.9735, Adjusted R-squared: 0.9721   
## Wald test: 662.2 on 2 and 36 DF, p-value: < 2.2e-16   
##   
## Diagnostic tests:  
## NULL  
##   
## alpha beta phi  
## Geometric coefficients: -0.8692494 3.276406 0.6881261

AIC(model.koyck)

## [1] -100.3319

BIC(model.koyck)

## [1] -93.67769

Dari hasil tersebut, didapat bahwa peubah dan memiliki nilai . Hal ini menunjukkan bahwa peubah dan berpengaruh signifikan terhadap . Adapun model keseluruhannya adalah sebagai berikut

### Peramalan dan Akurasi

Berikut adalah hasil peramalan y untuk 20 periode kedepan menggunakan model koyck

fore.koyck <- forecast(model = model.koyck, x=test$Xt, h=20)  
fore.koyck

## $forecasts  
## [1] 1.461356 1.471850 1.488502 1.509610 1.534226 1.561357 1.590169 1.620028  
## [9] 1.650683 1.682147 1.714376 1.746887 1.779170 1.810957 1.842098 1.872435  
## [17] 1.901847 1.930400 1.957667 1.982797  
##   
## $call  
## forecast.koyckDlm(model = model.koyck, x = test$Xt, h = 20)  
##   
## attr(,"class")  
## [1] "forecast.koyckDlm" "dLagM"

mape.koyck <- MAPE(fore.koyck$forecasts, test$Yt) #akurasi data test  
  
#akurasi data training  
GoF(model.koyck)

## n MAE MPE MAPE sMAPE MASE  
## model.koyck 39 0.04266313 0.0003653983 0.05629828 0.05652327 0.8258805  
## MSE MRAE GMRAE  
## model.koyck 0.003640555 2.032364 0.8127766

# Regression with Distributed Lag

Pemodelan model Regression with Distributed Lag dengan R dapat menggunakan dLagM::dlm() . Fungsi umum dari dlm adalah sebagai berikut.

dlm(formula , data , x , y , q , remove )

Fungsi dlm() akan menerapkan model lag terdistribusi dengan satu atau lebih prediktor. Nilai x dan y tidak perlu sebagai objek *time series* (ts). adalah integer yang mewakili panjang *lag* yang terbatas.

### Pemodelan (Lag=2)

model.dlm <- dlm(x = train$Xt,y = train$Yt , q = 2)  
summary(model.dlm)

##   
## Call:  
## lm(formula = model.formula, data = design)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.195814 -0.028795 0.004855 0.042836 0.158519   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.4607 0.1583 -2.910 0.00633 \*\*  
## x.t 32.0482 154.7816 0.207 0.83720   
## x.1 -186.9593 307.6174 -0.608 0.54738   
## x.2 165.3027 157.8285 1.047 0.30232   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.07915 on 34 degrees of freedom  
## Multiple R-squared: 0.9586, Adjusted R-squared: 0.9549   
## F-statistic: 262.2 on 3 and 34 DF, p-value: < 2.2e-16  
##   
## AIC and BIC values for the model:  
## AIC BIC  
## 1 -79.15833 -70.9704

AIC(model.dlm)

## [1] -79.15833

BIC(model.dlm)

## [1] -70.9704

Dari hasil diatas, didapat bahwa dari intercept<0.05$. Hal ini menunjukkan bahwa intercept berpengaruh signifikan terhadap . Adapun model keseluruhan yang terbentuk adalah sebagai berikut

### Peramalan dan Akurasi

Berikut merupakan hasil peramalan untuk 20 periode kedepan

fore.dlm <- forecast(model = model.dlm, x=test$Xt, h=20)  
fore.dlm

## $forecasts  
## [1] 1.412719 1.447680 1.480981 1.526389 1.550349 1.561023 1.587791 1.621388  
## [9] 1.659479 1.690253 1.712059 1.732674 1.773639 1.823095 1.867495 1.908877  
## [17] 1.951490 1.995244 2.026002 2.072968  
##   
## $call  
## forecast.dlm(model = model.dlm, x = test$Xt, h = 20)  
##   
## attr(,"class")  
## [1] "forecast.dlm" "dLagM"

mape.dlm <- MAPE(fore.dlm$forecasts, test$Yt)  
#akurasi data training  
GoF(model.dlm)

## n MAE MPE MAPE sMAPE MASE MSE  
## model.dlm 38 0.05681213 -0.006232561 0.08998629 0.08932243 1.076575 0.005604704  
## MRAE GMRAE  
## model.dlm 3.338191 1.228975

### *Lag* Optimum

#penentuan lag optimum   
finiteDLMauto(formula = Yt ~ Xt,  
 data = data.frame(train), q.min = 1, q.max = 20,  
 model.type = "dlm", error.type = "AIC", trace = TRUE)

## q - k MASE AIC BIC GMRAE MBRAE R.Adj.Sq Ljung-Box  
## 19 19 0.00000 -Inf -Inf 0.00000 0.00000 NaN NaN  
## 20 20 0.00000 -Inf -Inf 0.00000 0.00000 NaN NaN  
## 17 17 0.19764 -84.00666 -61.29678 0.26491 0.29241 0.98056 0.0002468295  
## 1 1 1.11285 -82.37309 -75.71884 1.17192 1.04384 0.95573 0.0001732082  
## 2 2 1.07658 -79.15833 -70.97040 1.22898 -0.81653 0.95492 0.0003533434  
## 18 18 0.18938 -78.67426 -55.76237 0.22001 0.25262 0.96592 0.0001477609  
## 3 3 1.02469 -76.44584 -66.78033 1.15799 1.11003 0.95439 0.0008943621  
## 4 4 0.96650 -76.35147 -65.26683 1.00834 -0.09290 0.95702 0.0099679065  
## 12 12 0.44311 -75.79342 -55.81036 0.61219 0.62611 0.97217 0.2363032112  
## 5 5 0.91014 -74.83528 -62.39249 1.02626 0.61059 0.95756 0.0228003972  
## 6 6 0.84973 -73.32246 -59.58521 0.95245 15.87705 0.95788 0.0677213971  
## 11 11 0.52964 -72.51794 -53.37580 0.56672 0.96516 0.96705 0.6459208959  
## 13 13 0.41520 -70.36862 -49.63523 0.44268 0.60106 0.96736 0.1225544606  
## 7 7 0.85249 -70.27857 -55.31349 1.11529 1.03842 0.95614 0.0842575628  
## 14 14 0.39502 -67.98133 -46.59369 0.47903 1.87246 0.96466 0.0723757005  
## 15 15 0.38375 -66.20011 -44.26034 0.47460 0.69445 0.96172 0.0034214215  
## 8 8 0.86368 -65.43721 -49.31411 1.21412 0.33227 0.95173 0.1747262256  
## 16 16 0.35449 -64.00123 -41.61820 0.41963 0.41262 0.95762 0.0726084306  
## 9 9 0.81242 -63.23638 -46.02854 1.13300 0.89529 0.95062 0.3523737533  
## 10 10 0.73278 -60.43275 -42.21719 0.82005 1.88186 0.94808 0.9184627744

Berdasarkan output tersebut, lag optimum didapatkan ketika lag=17. Selanjutnya dilakukan pemodelan untuk lag=17

#model dlm dengan lag optimum  
model.dlm2 <- dlm(x = train$Xt,y = train$Yt , q = 17)  
summary(model.dlm2)

##   
## Call:  
## lm(formula = model.formula, data = design)  
##   
## Residuals:  
## 1 2 3 4 5 6 7   
## -0.0005632 0.0034041 -0.0019066 0.0077268 -0.0221125 0.0028751 0.0240614   
## 8 9 10 11 12 13 14   
## -0.0258999 0.0210671 -0.0005691 -0.0086815 -0.0074026 0.0152597 -0.0174690   
## 15 16 17 18 19 20 21   
## 0.0162867 -0.0135370 0.0148733 -0.0076395 0.0030923 -0.0084815 0.0258737   
## 22 23   
## -0.0376498 0.0173918   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -2.098 4.170 -0.503 0.6413   
## x.t -63.154 911.888 -0.069 0.9481   
## x.1 421.848 2165.264 0.195 0.8550   
## x.2 -415.942 3090.429 -0.135 0.8994   
## x.3 1329.325 3462.207 0.384 0.7206   
## x.4 -4032.940 3339.095 -1.208 0.2937   
## x.5 6295.706 3152.815 1.997 0.1165   
## x.6 -6502.392 2918.478 -2.228 0.0898 .  
## x.7 2920.883 2756.830 1.060 0.3491   
## x.8 2221.282 2829.060 0.785 0.4763   
## x.9 -4655.388 3099.354 -1.502 0.2075   
## x.10 5215.444 3449.439 1.512 0.2051   
## x.11 -5870.290 3501.135 -1.677 0.1689   
## x.12 6227.603 2882.007 2.161 0.0968 .  
## x.13 -3874.550 1924.970 -2.013 0.1144   
## x.14 -198.117 846.797 -0.234 0.8265   
## x.15 1653.312 602.973 2.742 0.0518 .  
## x.16 -997.940 409.088 -2.439 0.0713 .  
## x.17 342.647 163.464 2.096 0.1041   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.03916 on 4 degrees of freedom  
## Multiple R-squared: 0.9965, Adjusted R-squared: 0.9806   
## F-statistic: 62.65 on 18 and 4 DF, p-value: 0.0005517  
##   
## AIC and BIC values for the model:  
## AIC BIC  
## 1 -84.00666 -61.29678

AIC(model.dlm2)

## [1] -84.00666

BIC(model.dlm2)

## [1] -61.29678

Dari hasil tersebut tidak ada peubah yang berpengaruh signifikan terhadap taraf nyata 5% tetapi , , , berpengaruh signifikan terhadap taraf nyata 10%. Adapun keseluruhan model yang terbentuk adalah

Adapun hasil peramalan 20 periode kedepan menggunakan model tersebut adalah sebagai berikut

#peramalan dan akurasi  
fore.dlm2 <- forecast(model = model.dlm2, x=test$Xt, h=20)  
mape.dlm2<- MAPE(fore.dlm2$forecasts, test$Yt)  
#akurasi data training  
GoF(model.dlm2)

## n MAE MPE MAPE sMAPE MASE  
## model.dlm2 23 0.01320975 -0.0002149902 0.01410767 0.01409464 0.1976422  
## MSE MRAE GMRAE  
## model.dlm2 0.0002666773 0.9745025 0.2649094

Model tersebut merupakan model yang sangat baik dengan nilai MAPE yang kurang dari 10%.

# Model Autoregressive

Peubah dependen dipengaruhi oleh peubah independen pada waktu sekarang, serta dipengaruhi juga oleh peubah dependen itu sendiri pada satu waktu yang lalu maka model tersebut disebut *autoregressive* (Gujarati 2004).

### Pemodelan

Pemodelan Autoregressive dilakukan menggunakan fungsi dLagM::ardlDlm() . Fungsi tersebut akan menerapkan *autoregressive* berordo dengan satu prediktor. Fungsi umum dari ardlDlm() adalah sebagai berikut.

ardlDlm(formula = NULL , data = NULL , x = NULL , y = NULL , p = 1 , q = 1 ,   
 remove = NULL )

Dengan adalah integer yang mewakili panjang *lag* yang terbatas dan adalah integer yang merepresentasikan ordo dari proses *autoregressive*.

#model.ardl <- ardlDlm(x = train$Xt, y = train$Yt, p = 1 , q = 1)  
#summary(model.ardl)  
#AIC(model.ardl)  
#BIC(model.ardl)

model.ardl <- ardlDlm(formula = Yt ~ Xt,   
 data = train,p = 1 , q = 1)  
summary(model.ardl)

##   
## Time series regression with "ts" data:  
## Start = 2, End = 40  
##   
## Call:  
## dynlm(formula = as.formula(model.text), data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.175661 -0.019623 0.002162 0.015839 0.132589   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.1668 0.1295 -1.288 0.206   
## Xt.t -52.6597 44.0007 -1.197 0.239   
## Xt.1 56.7052 44.6207 1.271 0.212   
## Yt.1 0.6262 0.1305 4.799 2.94e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.06227 on 35 degrees of freedom  
## Multiple R-squared: 0.9747, Adjusted R-squared: 0.9725   
## F-statistic: 449.5 on 3 and 35 DF, p-value: < 2.2e-16

AIC(model.ardl)

## [1] -100.0924

BIC(model.ardl)

## [1] -91.77457

Hasil di atas menunjukkan bahwa selain peubah , hasil uji t menunjukkan nilai-p pada peubah Hal ini menunjukkan bahwa peubah berpengaruh signifikan terhadap , sementara dan berpengaruh signifikan terhadap . Model keseluruhannya adalah sebagai berikut:

### Peramalan dan Akurasi

fore.ardl <- forecast(model = model.ardl, x=test$Xt, h=20)  
fore.ardl

## $forecasts  
## [1] 1.476392 1.500268 1.531450 1.559126 1.581261 1.605965 1.634816 1.667165  
## [9] 1.698589 1.726566 1.753528 1.787398 1.828159 1.871348 1.915146 1.959749  
## [17] 2.004684 2.046924 2.094805 2.154326  
##   
## $call  
## forecast.ardlDlm(model = model.ardl, x = test$Xt, h = 20)  
##   
## attr(,"class")  
## [1] "forecast.ardlDlm" "dLagM"

Data di atas merupakan hasil peramalan untuk 20 periode ke depan menggunakan Model Autoregressive dengan dan .

mape.ardl <- MAPE(fore.ardl$forecasts, test$Yt)  
mape.ardl

## [1] 0.06675562

#akurasi data training  
GoF(model.ardl)

## n MAE MPE MAPE sMAPE MASE MSE  
## model.ardl 39 0.0405306 -0.004605892 0.05433182 0.0539828 0.7845987 0.003479876  
## MRAE GMRAE  
## model.ardl 2.055801 0.6817497

Berdasarkan akurasi di atas, terlihat bahwa nilai MAPE keduanya tidak jauh berbeda. Artinya, model regresi dengan distribusi lag ini tidak overfitted atau underfitted

### *Lag* Optimum

#penentuan lag optimum  
model.ardl.opt <- ardlBoundOrders(data = data.frame(data), ic = "AIC",   
 formula = Yt ~ Xt )  
min\_p=c()  
for(i in 1:15){  
 min\_p[i]=min(model.ardl.opt$Stat.table[[i]])  
}  
q\_opt=which(min\_p==min(min\_p, na.rm = TRUE))  
p\_opt=which(model.ardl.opt$Stat.table[[q\_opt]] ==   
 min(model.ardl.opt$Stat.table[[q\_opt]], na.rm = TRUE))  
data.frame("q\_optimum" = q\_opt, "p\_optimum" = p\_opt,   
 "AIC"=model.ardl.opt$min.Stat)

## q\_optimum p\_optimum AIC  
## 1 1 1 -123.6981

Dari tabel di atas, dapat terlihat bahwa nilai AIC terendah didapat ketika dan , yaitu sebesar -123.6981. Artinya, model autoregressive optimum didapat ketika dan .

Selanjutnya dapat dilakukan pemodelan dengan nilai dan optimum seperti inisialisasi di langkah sebelumnya.

# Pemodelan DLM & ARDL dengan Library dynlm

Pemodelan regresi dengan peubah *lag* tidak hanya dapat dilakukan dengan fungsi pada *packages* dLagM , tetapi terdapat *packages* dynlm yang dapat digunakan. Fungsi dynlm secara umum adalah sebagai berikut.

dynlm(formula, data, subset, weights, na.action, method = "qr",  
 model = TRUE, x = FALSE, y = FALSE, qr = TRUE, singular.ok = TRUE,  
 contrasts = NULL, offset, start = NULL, end = NULL, ...)

Untuk menentukan formula model yang akan digunakan, tersedia fungsi tambahan yang memungkinkan spesifikasi dinamika (melalui d() dan L()) atau pola linier/siklus dengan mudah (melalui trend(), season(), dan harmon()). Semua fungsi formula baru mengharuskan argumennya berupa objek deret waktu (yaitu, "ts" atau "zoo").

#sama dengan model dlm q=1  
cons\_lm1 <- dynlm(Yt ~ Xt+L(Xt),data = train.ts)  
#sama dengan model ardl p=1 q=0  
cons\_lm2 <- dynlm(Yt ~ Xt+L(Yt),data = train.ts)  
#sama dengan ardl p=1 q=1  
cons\_lm3 <- dynlm(Yt ~ Xt+L(Xt)+L(Yt),data = train.ts)  
#sama dengan dlm p=2  
cons\_lm4 <- dynlm(Yt ~ Xt+L(Xt)+L(Xt,2),data = train.ts)

### Ringkasan Model

summary(cons\_lm1)

##   
## Time series regression with "ts" data:  
## Start = 2, End = 40  
##   
## Call:  
## dynlm(formula = Yt ~ Xt + L(Xt), data = train.ts)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.193317 -0.034997 -0.001616 0.045411 0.169691   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.4246 0.1496 -2.838 0.00741 \*\*  
## Xt -131.5296 51.8216 -2.538 0.01561 \*   
## L(Xt) 141.6732 52.0012 2.724 0.00988 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.07906 on 36 degrees of freedom  
## Multiple R-squared: 0.9581, Adjusted R-squared: 0.9557   
## F-statistic: 411.2 on 2 and 36 DF, p-value: < 2.2e-16

summary(cons\_lm2)

##   
## Time series regression with "ts" data:  
## Start = 2, End = 40  
##   
## Call:  
## dynlm(formula = Yt ~ Xt + L(Yt), data = train.ts)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.182019 -0.025875 -0.002459 0.024856 0.147801   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.2678 0.1031 -2.597 0.01355 \*   
## Xt 3.2388 1.1511 2.814 0.00789 \*\*   
## L(Yt) 0.6920 0.1208 5.729 1.6e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.0628 on 36 degrees of freedom  
## Multiple R-squared: 0.9735, Adjusted R-squared: 0.9721   
## F-statistic: 662.2 on 2 and 36 DF, p-value: < 2.2e-16

summary(cons\_lm3)

##   
## Time series regression with "ts" data:  
## Start = 2, End = 40  
##   
## Call:  
## dynlm(formula = Yt ~ Xt + L(Xt) + L(Yt), data = train.ts)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.175661 -0.019623 0.002162 0.015839 0.132589   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.1668 0.1295 -1.288 0.206   
## Xt -52.6597 44.0007 -1.197 0.239   
## L(Xt) 56.7052 44.6207 1.271 0.212   
## L(Yt) 0.6262 0.1305 4.799 2.94e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.06227 on 35 degrees of freedom  
## Multiple R-squared: 0.9747, Adjusted R-squared: 0.9725   
## F-statistic: 449.5 on 3 and 35 DF, p-value: < 2.2e-16

summary(cons\_lm4)

##   
## Time series regression with "ts" data:  
## Start = 3, End = 40  
##   
## Call:  
## dynlm(formula = Yt ~ Xt + L(Xt) + L(Xt, 2), data = train.ts)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.195814 -0.028795 0.004855 0.042836 0.158519   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.4607 0.1583 -2.910 0.00633 \*\*  
## Xt 32.0482 154.7816 0.207 0.83720   
## L(Xt) -186.9593 307.6174 -0.608 0.54738   
## L(Xt, 2) 165.3027 157.8285 1.047 0.30232   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.07915 on 34 degrees of freedom  
## Multiple R-squared: 0.9586, Adjusted R-squared: 0.9549   
## F-statistic: 262.2 on 3 and 34 DF, p-value: < 2.2e-16

### SSE

deviance(cons\_lm1)

## [1] 0.2250182

deviance(cons\_lm2)

## [1] 0.1419774

deviance(cons\_lm3)

## [1] 0.1357152

deviance(cons\_lm4)

## [1] 0.2129788

### Uji Diagnostik

#uji model  
if(require("lmtest")) encomptest(cons\_lm1, cons\_lm2)

## Encompassing test  
##   
## Model 1: Yt ~ Xt + L(Xt)  
## Model 2: Yt ~ Xt + L(Yt)  
## Model E: Yt ~ Xt + L(Xt) + L(Yt)  
## Res.Df Df F Pr(>F)   
## M1 vs. ME 35 -1 23.031 2.94e-05 \*\*\*  
## M2 vs. ME 35 -1 1.615 0.2122   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#### Autokorelasi

#durbin watson  
dwtest(cons\_lm1)

##   
## Durbin-Watson test  
##   
## data: cons\_lm1  
## DW = 0.78629, p-value = 1.555e-06  
## alternative hypothesis: true autocorrelation is greater than 0

dwtest(cons\_lm2)

##   
## Durbin-Watson test  
##   
## data: cons\_lm2  
## DW = 2.3662, p-value = 0.8069  
## alternative hypothesis: true autocorrelation is greater than 0

dwtest(cons\_lm3)

##   
## Durbin-Watson test  
##   
## data: cons\_lm3  
## DW = 2.31, p-value = 0.6979  
## alternative hypothesis: true autocorrelation is greater than 0

dwtest(cons\_lm4)

##   
## Durbin-Watson test  
##   
## data: cons\_lm4  
## DW = 0.84146, p-value = 5.139e-06  
## alternative hypothesis: true autocorrelation is greater than 0

#### Heterogenitas

bptest(cons\_lm1)

##   
## studentized Breusch-Pagan test  
##   
## data: cons\_lm1  
## BP = 4.6087, df = 2, p-value = 0.09983

bptest(cons\_lm2)

##   
## studentized Breusch-Pagan test  
##   
## data: cons\_lm2  
## BP = 14.629, df = 2, p-value = 0.0006659

bptest(cons\_lm3)

##   
## studentized Breusch-Pagan test  
##   
## data: cons\_lm3  
## BP = 13.942, df = 3, p-value = 0.002985

bptest(cons\_lm4)

##   
## studentized Breusch-Pagan test  
##   
## data: cons\_lm4  
## BP = 3.8094, df = 3, p-value = 0.2828

#### Kenormalan

shapiro.test(residuals(cons\_lm1))

##   
## Shapiro-Wilk normality test  
##   
## data: residuals(cons\_lm1)  
## W = 0.98083, p-value = 0.7337

shapiro.test(residuals(cons\_lm2))

##   
## Shapiro-Wilk normality test  
##   
## data: residuals(cons\_lm2)  
## W = 0.95759, p-value = 0.1481

shapiro.test(residuals(cons\_lm3))

##   
## Shapiro-Wilk normality test  
##   
## data: residuals(cons\_lm3)  
## W = 0.94689, p-value = 0.06457

shapiro.test(residuals(cons\_lm4))

##   
## Shapiro-Wilk normality test  
##   
## data: residuals(cons\_lm4)  
## W = 0.98095, p-value = 0.7509

# Kesimpulan

## Perbandingan Model

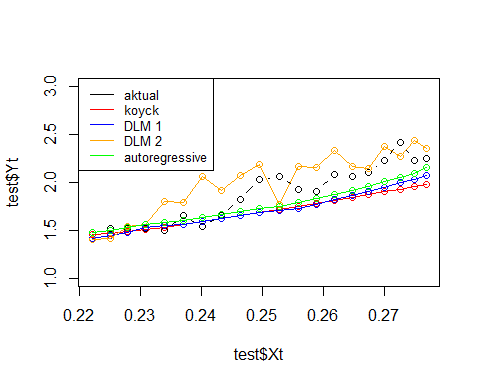
akurasi <- matrix(c(mape.koyck, mape.dlm, mape.dlm2, mape.ardl))  
row.names(akurasi)<- c("Koyck","DLM 1","DLM 2","Autoregressive")  
colnames(akurasi) <- c("MAPE")  
akurasi

## MAPE  
## Koyck 0.08834523  
## DLM 1 0.08035993  
## DLM 2 0.09873475  
## Autoregressive 0.06675562

Berdasarkan nilai MAPE, model paling optimum didapat pada Model Autoregressive karena memiliki nilai MAPE yang terkecil.

### Plot

par(mfrow=c(1,1))  
plot(test$Xt, test$Yt, type="b", col="black", ylim=c(1,3))  
points(test$Xt, fore.koyck$forecasts,col="red")  
lines(test$Xt, fore.koyck$forecasts,col="red")  
points(test$Xt, fore.dlm$forecasts,col="blue")  
lines(test$Xt, fore.dlm$forecasts,col="blue")  
points(test$Xt, fore.dlm2$forecasts,col="orange")  
lines(test$Xt, fore.dlm2$forecasts,col="orange")  
points(test$Xt, fore.ardl$forecasts,col="green")  
lines(test$Xt, fore.ardl$forecasts,col="green")  
legend("topleft",c("aktual", "koyck","DLM 1","DLM 2", "autoregressive"), lty=1, col=c("black","red","blue","orange","green"), cex=0.8)



Berdasarkan plot tersebut, terlihat bahwa plot yang paling mendekati data aktualnya adalah Model Autoregressive, sehingga dapat disimpulkan model terbaik dalam hal ini adalah model regresi Autoregressive

par(mfrow=c(1,1))  
plot(test$Xt, test$Yt, type="b", col="black", ylim=c(1,3),main="Aktual vs Autoregressive")  
par(mfrow=c(1,1))  
plot(test$Xt, test$Yt, type="b", col="black", ylim=c(1,3),main="Aktual vs Autoregressive")  
points(test$Xt, fore.koyck$forecasts,col="red")  
lines(test$Xt, fore.koyck$forecasts,col="red")

