Module11HW

Diana McSpadden

11/11/2020

Stat 6021: Homework Set 11

H Diana McSpadden (hdm5s)

Attended Study Group With: Nam, Caprill, Barbre, Kastanowski, and Bernhardt

1. The data set “mcgill.txt” contains the seasonally adjusted quarterly sales for the McGill Company (response variable, first column, in million dollars) and for the entire industry (predictor, second column, in million dollars).

(a) Explain why fitting a simple linear regression model with autocorrelated errors is a better choice than a simple linear regression model with i.i.d. errors for this data set.

Answer Q1a:

Due to the residuals not meeting the SLR assumption of lack of correlation, if we fit without taking account of these correlated errors : 1. coefficients will not be minimum variance estimates. 2. when errors are positively autocorrelated the MSE seriously underestimates the error variance: sigma^2 resulting in: * standard errors of coefficients may be too small, and CIs will be too small 3. and CIs, PIs, hypothesis tests on t or F values are no longer exact.

If we fix the issue of autocorrelation of errors (and the other assumptions are met) we will not have the issues above.

(b) Use the Cochrane-Orcutt method to fit a simple linear regression model with AR(1) errors.

```
#setup
#library(dplyr)

# get the data
mcData<-read.table("mcgill.txt", header=FALSE, sep="")
```

```
colnames(mcData)<-c("company","industry")
mcData
```

```
##    company industry
## 1    20.96    127.3
## 2    21.40    130.0
## 3    21.96    132.7
## 4    21.52    129.4
## 5    22.39    135.0
## 6    22.76    137.1
## 7    23.48    141.1
## 8    23.66    142.8
## 9    24.10    145.5
## 10   24.01    145.3
## 11   24.54    148.3
## 12   24.28    146.4
## 13   25.00    150.2
## 14   25.64    153.1
## 15   26.46    157.3
## 16   26.98    160.7
## 17   27.52    164.2
## 18   27.78    165.6
## 19   28.24    168.7
## 20   28.78    172.0
```

```
attach(mcData)

?arima
```

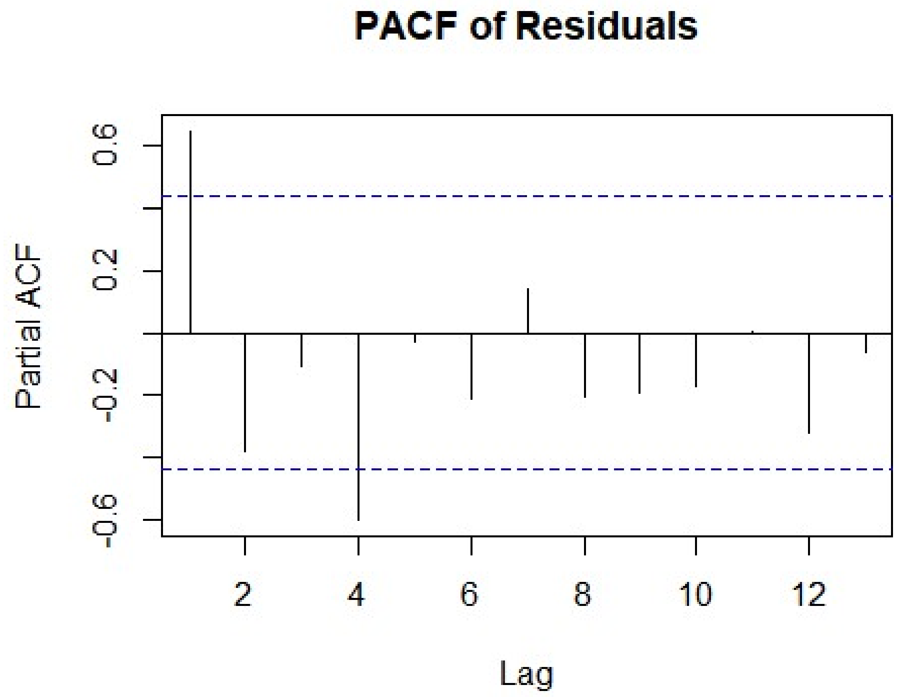
```
## starting httpd help server ... done
```

```
mcResult<-lm(company~industry)
summary(mcResult)
```

```
## 
## Call:
## lm(formula = company ~ industry)
## 
## Residuals:
##       Min        1Q    Median        3Q       Max 
## -0.151659 -0.068633 -0.003432  0.046715  0.184384 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -1.434832   0.241956   -5.93  1.3e-05 ***
## industry     0.176163   0.001632  107.93  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.09744 on 18 degrees of freedom
## Multiple R-squared:  0.9985, Adjusted R-squared:  0.9984
## F-statistic: 1.165e+04 on 1 and 18 DF,  p-value: < 2.2e-16
```

```
mcRes<-mcResult$residuals

##PACF plot of residuals
pacf(mcRes, main="PACF of Residuals")
```



```
# significant at lag 1


##fit an AR(1) model for residuals
# order = c(1,0,0) indicates the p value == 1
ar.1<-arima(mcRes, order = c(1,0,0), include.mean = FALSE)
ar.1
```

```
##
## Call:
## arima(x = mcRes, order = c(1, 0, 0), include.mean = FALSE)
##
## Coefficients:
##          ar1
##       0.6442
## s.e.  0.1627
##
## sigma^2 estimated as 0.004827:  log likelihood = 24.69,  aic = -45.38
```

```
ar.1$coef

##       ar1
## 0.6442108
```

This looks significant at lag 1 **and** 4.

i. Report the estimated autocorrelation at lag 1 for the errors.

Answer Q1bi: 0.6442

ii. Write out the model with the estimated values of the coefficients.

y’ = (-1.43)(1 - 0.6442) + (0.176)(x(t) - (0.6442)x(t-1)) + E(t) - 0.6442E(t-1)

Answer Q1bii:

y’ = (0.509) + 0.176(x’) + a(t)

iii. Assess if the regression model assumptions are met.

First, transform company and industry.

```
##transform response and predictor
shift<-ar.1$coef

#?lag

#create new column bind, shifts company variable by 1
y <- cbind(as.ts(company),lag(company))
yprime <- y[,2] - shift*y[,1]
y

## Time Series:
## Start = 0
## End = 20
## Frequency = 1
##    as.ts(company) lag(company)
##  0             NA        20.96
##  1          20.96        21.40
##  2          21.40        21.96
##  3          21.96        21.52
##  4          21.52        22.39
##  5          22.39        22.76
##  6          22.76        23.48
##  7          23.48        23.66
##  8          23.66        24.10
##  9          24.10        24.01
## 10          24.01        24.54
## 11          24.54        24.28
## 12          24.28        25.00
## 13          25.00        25.64
## 14          25.64        26.46
## 15          26.46        26.98
## 16          26.98        27.52
## 17          27.52        27.78
## 18          27.78        28.24
## 19          28.24        28.78
## 20          28.78           NA
```

```
yprime
```

```
## Time Series:
## Start = 0
## End = 20
## Frequency = 1
##  [1]        NA  7.897341  8.173889  7.373131  8.526583  8.336120  8.817762
##  [8]  8.533930  8.857972  8.484519  9.072498  8.471067  9.358561  9.534730
## [15]  9.942435  9.934182 10.139192 10.051318 10.343824 10.587487        NA
```

```
x<-cbind(as.ts(industry),lag(industry))
xprime<-x[,2] - shift*x[,1]
x
```

```
## Time Series:
## Start = 0
## End = 20
## Frequency = 1
##    as.ts(industry) lag(industry)
##  0              NA         127.3
##  1           127.3         130.0
##  2           130.0         132.7
##  3           132.7         129.4
##  4           129.4         135.0
##  5           135.0         137.1
##  6           137.1         141.1
##  7           141.1         142.8
##  8           142.8         145.5
##  9           145.5         145.3
## 10           145.3         148.3
## 11           148.3         146.4
## 12           146.4         150.2
## 13           150.2         153.1
## 14           153.1         157.3
## 15           157.3         160.7
## 16           160.7         164.2
## 17           164.2         165.6
## 18           165.6         168.7
## 19           168.7         172.0
## 20           172.0            NA
```

```
xprime
```

```
## Time Series:
## Start = 0
## End = 20
## Frequency = 1
##  [1]       NA 47.99196 48.95259 43.91323 51.63912 50.13154 52.77870
51.90185
##  [9] 53.50670 51.56733 54.69617 50.86354 55.88754 56.33954 58.67132
59.36564
## [17] 60.67532 59.82058 62.01869 63.32164       NA
```

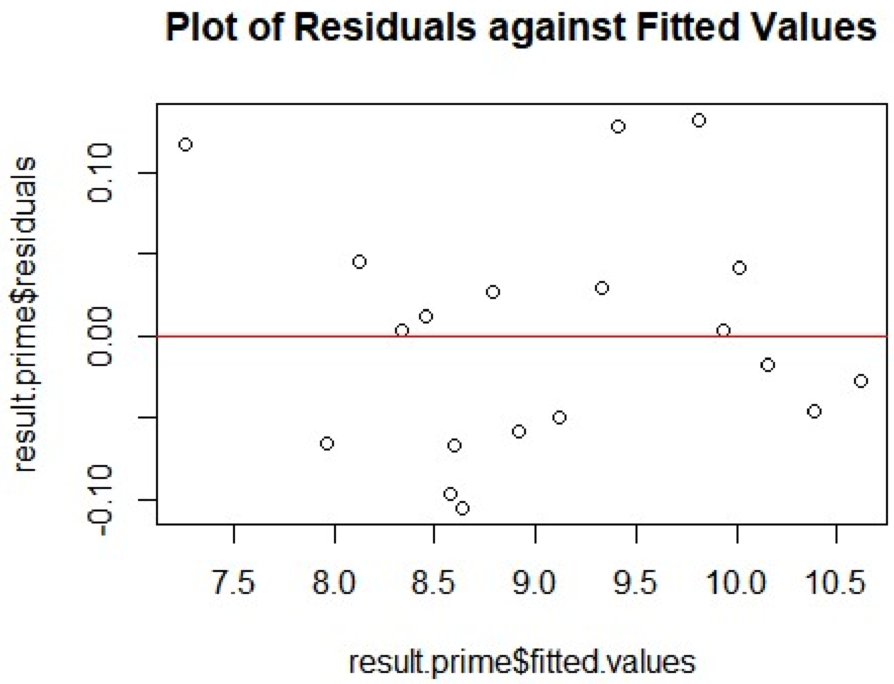
```r
##perform regression on transformed variables
result.prime<-lm(yprime~xprime)

result.prime
```

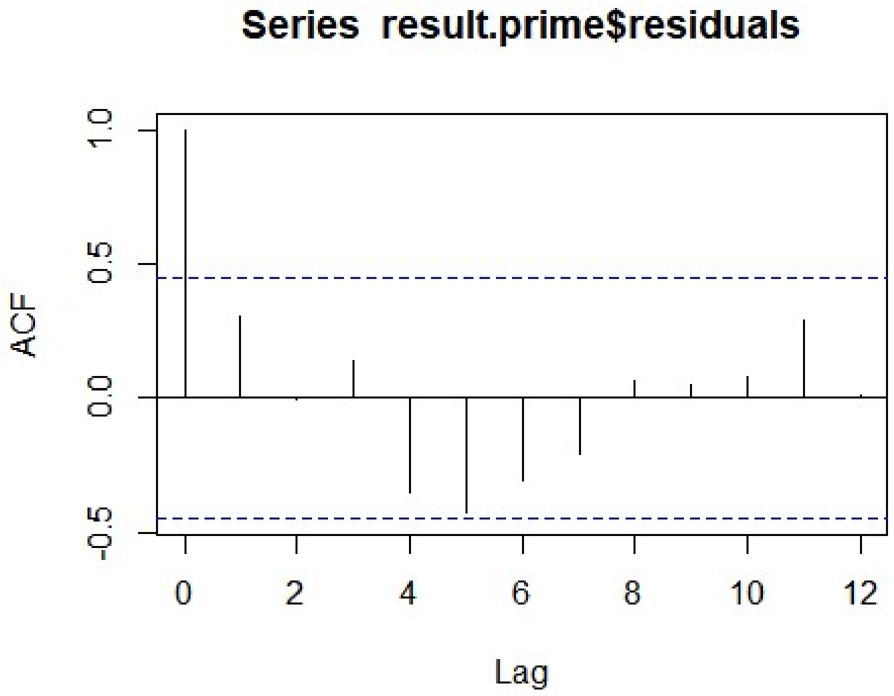
```
##
## Call:
## lm(formula = yprime ~ xprime)
##
## Coefficients:
## (Intercept)       xprime
##     -0.3421       0.1730
```

Second, now test the SLR assumptions:

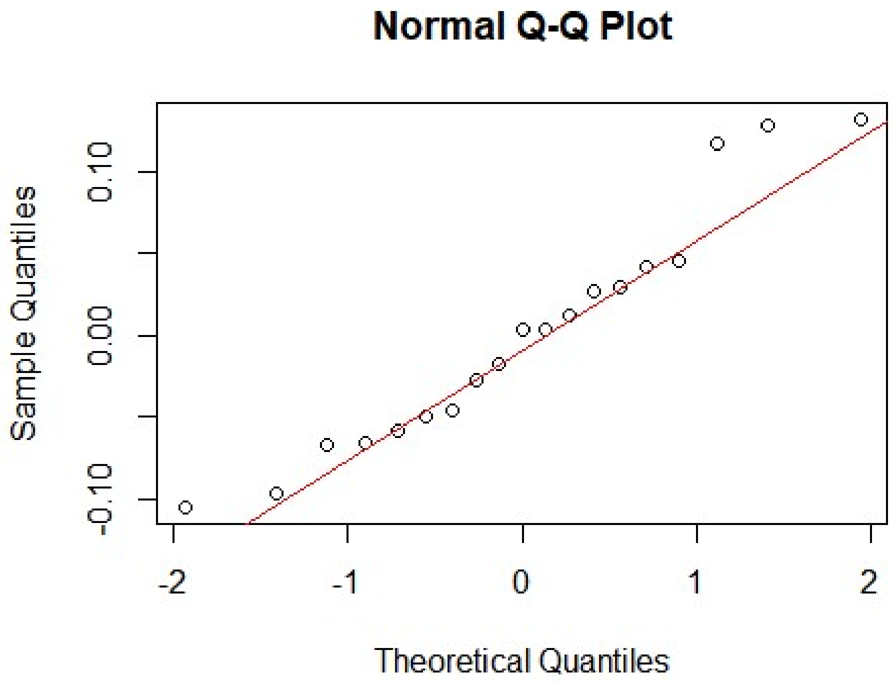
```r
##residual plot
plot(result.prime$fitted.values,result.prime$residuals, main="Plot of
Residuals against Fitted Values")
abline(h=0,col="red")
```



```r
##acf plot of residuals
acf(result.prime$residuals)
```



```
##qq plot of residuals
qqnorm(result.prime$residuals)
qqline(result.prime$residuals, col="red")
```



Answer Q1biii:

Yes, based on the residual plot, ACF plot, and QQ plot, the linear regression assumptions are met.

iv. Are the seasonally adjusted quarterly sales for the McGill Company significantly linearly related to the seasonally adjusted quarterly sales for the entireindustry?

Be sure to state the hypothesis statements, test statistic, and p-value, as well as an appropriate conclusion in context.

```
summary(result.prime)

##
## Call:
## lm(formula = yprime ~ xprime)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.105352 -0.054672  0.003181  0.035661  0.131732
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.342054   0.180445  -1.896   0.0751 .
## xprime       0.173045   0.003301  52.419   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07323 on 17 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.9939, Adjusted R-squared:  0.9935
## F-statistic:  2748 on 1 and 17 DF,  p-value: < 2.2e-16
```

Answer q1biv:

H0: B1 == 0 Ha: B1 != 0, where B1 is the coefficient for xprime.

Yes, these are linearly related because the t-statistic is large (52.42), and the p-value (2e-16) is less than 0.05. We reject the null hypothesis and conclude that for every 1 million in industry sales mcgills sales increase by $173,000.

2. Show how to apply the Cochrane-Orcutt method to a multiple linear regression model …

with AR(1) errors, i.e. what kind of transformations to the variables need to be made. Show how you derived your answer.

Answer Q2:

Each variable, response and predictors, needs to be corrected by (phi * (variable at the correct lag)). So for each observation, i:

y(i)’ = y(i) - (phi * y(i-lag))

and each regressor x(n):

x(i,n)’ = x(i,n) - (phi * x(i-lag,n)), and in this case lag = 1.

3. Consider a applying the Cochrane Orcutt method, but now …

to a simple linear regression with AR(2) errors, where Et = phi1(t-1) + phi2(t-2) + at.

(a) Show how the Cochrane-Orcutt method should be applied, …

i.e. what kind of transformations to the variables need to be made. Show how you derived your answer.

Answer Q3a:

The response and predictor needs to be corrected by (phi(j) * (variable at the j lag)) for j = 1 to p. So, for lag == 2, for each observation, i:

The response variable:

y(i)’ = y(i) - (phi1 * y(i-1)) - (phi2 * y(i-2))

and the regressor:

x(i)’ = x(i) - (phi1 * x(i-1)) - (phi2 * x(i-2))

(b) What do you think are the transformations to the variables for a simple linear regression model with AR(p) errors, …

where p is a positive integer. You do not have to show how you arrived at your answer.

Answer Q3b:

The response and predictor needs to be corrected by (phi(j) * (variable at the j lag)) for j = 1 to p. So, for each observation, i:

The response variable:

y(i)’ = y(i) - (phi1 * y(i-1)) - … - (phip * y(i-p))

and the regressor:

x(i)’ = x(i) - (phi1 * x(i-1)) - … - (phip * x(i-p))

Hint: For questions 2 and 3, the derivation from the textbook page 482, equation (14.7) will be helpful. Please note the typo in the last line of (14.7), Et should be at.

4. Submit your group’s code and estimated test MSE from guided question set 11.

```
# setup
library(boot)

# get the data & attach
data<-read.table("nfl.txt", header=TRUE, sep="")
attach(data)

## The following object is masked _by_ .GlobalEnv:
##
##     y

# myLOOCVLoop function
# inpuut: the dataset: dataframe, leaveOutIndex: the index to use as the test
value
# output: the MSE for the requested loop of LOOCV
myLOOCVLoop <- function(theData, leaveOutIndex) {

  # leave out the i-th value for testing
  test <- theData[leaveOutIndex, ]
  # keep rest of data for training
  train <- theData[-leaveOutIndex, ]
```

```r
  # fit to training data
  glm.fit<-glm(y~x2+x7+x8, data=train)

  ## get predicted value from test data
  preds<-predict(glm.fit,newdata=test)

  #print(paste("Prediction: ", preds))
  #print(paste("Actual: ", test[1]))

  # calculate square of difference between predicted and actual
  difsquared = (preds - test[1])^2

  # return dif squared
  return(difsquared)
}

# I want to calculate: #mean((data$actual - data$pred)^2)
# create an accumulator variable
difSquaredAccummulator <- 0

# loop through all rows in data set
for (i in 1:nrow(data))
{
  # add to accumulator the dif Squared
  difSquaredAccummulator <- difSquaredAccummulator + myLOOCVLoop(data,i)
}

# get average of dif squared == MSE for LOOCV
mseLOOCV <- difSquaredAccummulator / nrow(data)

print(paste("My LOOCV: ", mseLOOCV))
```

```
## [1] "My LOOCV:  3.123615380779"
```

My test MSE: 3.123615380779

5. Please remember to complete the Module 11 Guided Question Set Participation Self-and Peer-Evaluation Questions via Test & Quizzes on Collab.

Will do