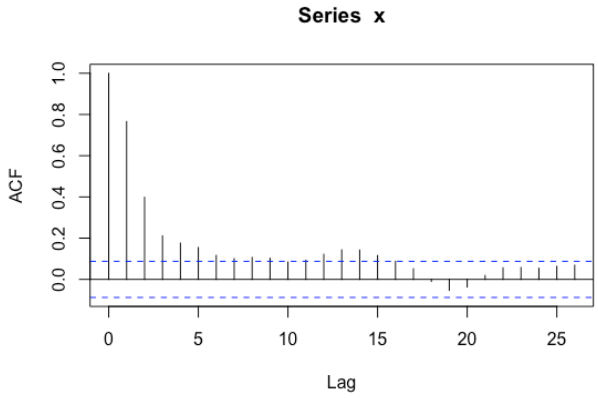
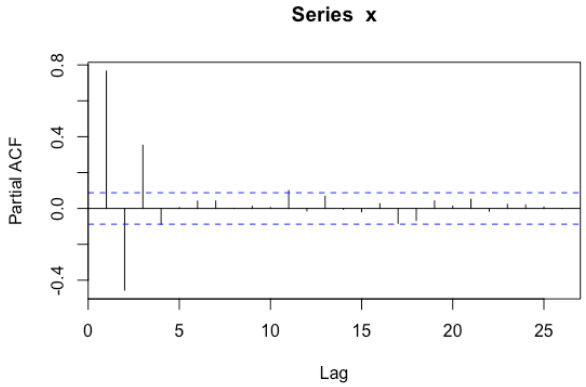
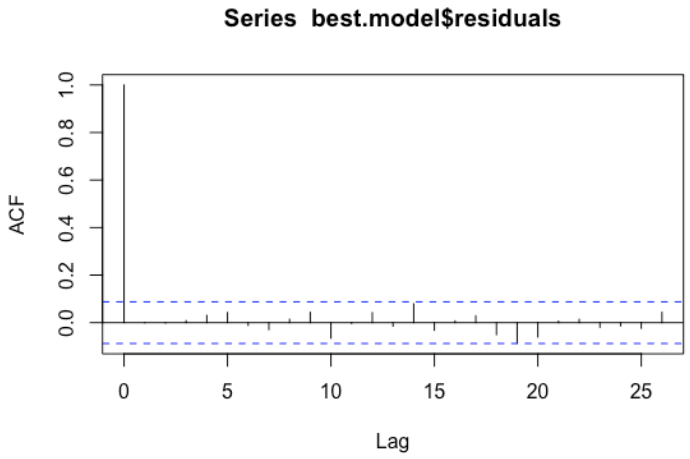
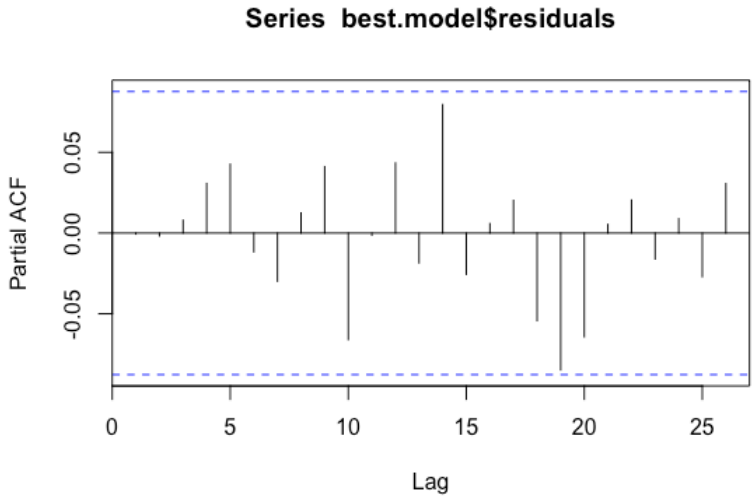
Part 1)
###ACF and PACF of x



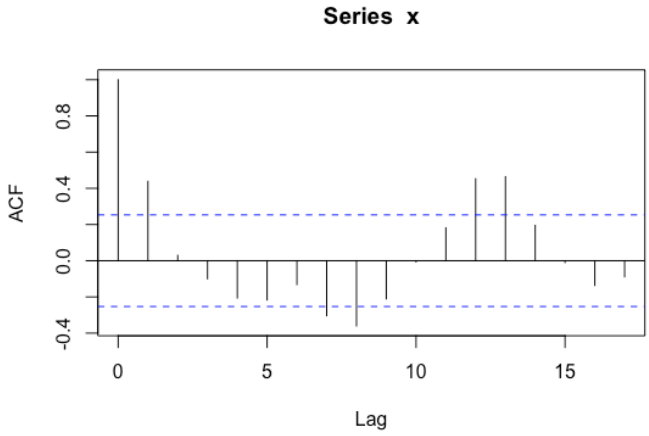


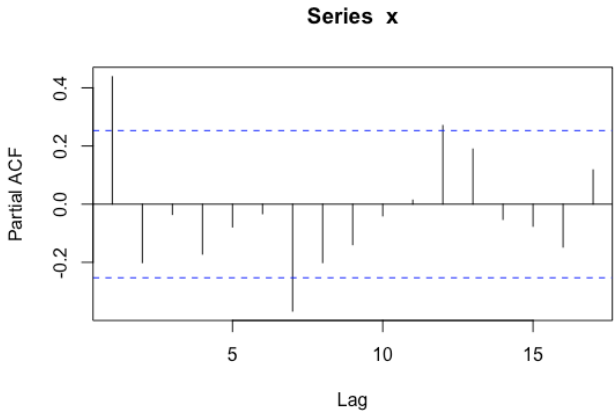
####acf and pacf of the residuals of the best model of ARIMA(p.best,d.best,q.best)





#part 2
###Acf and pacf of residuals of the least squares model





```
> stargazer(best.model,best.model2,best.model3,type="text")

================================================
                        Dependent variable:
                    ----------------------------
                                x
                       (1)             (2)             (3)
------------------------------------------------
ar1                  1.375***        0.988***        1.746***
                    (0.247)         (0.126)         (0.231)

ar2                  -0.896*         -0.578***       -1.399***
                    (0.519)         (0.180)         (0.495)

ar3                  -0.185          0.070           0.726
                    (0.589)         (0.194)         (0.502)

ar4                  0.305           -0.178          -0.413*
                    (0.403)         (0.189)         (0.237)

ar5                  -0.348          -0.113
                    (0.281)         (0.192)

ar6                  0.313           0.197
                    (0.230)         (0.176)

ar7                  -0.446***       -0.447***
                    (0.138)         (0.121)

ma1                  -1.543***       -1.000***       -2.028***
                    (0.258)         (0.056)         (0.237)

ma2                  0.657           1.146**
                    (0.567)         (0.523)

ma3                  0.594           -0.164
                    (0.561)         (0.519)

ma4                  -0.708***       0.097
                    (0.259)         (0.334)

ma5                                  0.222
                                    (0.329)

ma6                                  0.002
                                    (0.339)

ma7                                  -0.903***
                                    (0.333)

ma8                                  0.631***
                                    (0.173)
```

intercept	0.008***	0.007***	0.008***
	(0.002)	(0.002)	(0.001)

Observations	60	60	60
Log Likelihood	9.777	6.472	9.779
sigma2	0.033	0.041	0.031
Akaike Inf. Crit.	6.446	7.056	8.443
==========			
Note:	*p<0.1; **p<0.05; ***p<0.01		

All of the three best models with the lowest AIC were ARMA models, d=0.

APPENDIX:

PART 1:

```
x <- arima.sim(list(ar=c(0.7,-0.5,.3),
                ma=c(0.7,0.5,0.2)),n = 500)

####plot the acf and pacf of x
acf(x)
pacf(x)

max.order <- 10
AIC.matrix <- list()


###Fit a series of ARIMA models with 0 ≤ p ≤ 10, 0 ≤ q ≤ 10, 1 ≤ d ≤ 3,
##and store the AIC of each fit.
max.d <- 3
for(d in 0:max.d){
 AIC.temp.matrix <- matrix(0,nrow = max.order+1,ncol= max.order+1)
 for(i in 1:(max.order+1)){
   for(j in 1:(max.order+1)){
     currentArima <- arima(x,order=c(i-1,d,j-1))
     AIC.temp.matrix[i,j] <- AIC(currentArima)
   }
 }
 AIC.matrix[[d+1]] <- AIC.temp.matrix
}



###find the minimum aic of each AIC matrix, and store minimum aic of each matrix
d.0<-min(AIC.matrix[[1]]) ##ARIMA(0 ≤ p ≤ 10,0,0 ≤ q ≤ 10)
d.1<-min(AIC.matrix[[2]]) ##ARIMA(0 ≤ p ≤ 10,1,0 ≤ q ≤ 10)
d.2<-min(AIC.matrix[[3]]) ##ARIMA(0 ≤ p ≤ 10,2,0 ≤ q ≤ 10)
d.3<-min(AIC.matrix[[4]]) ##ARIMA(0 ≤ p ≤ 10,3,0 ≤ q ≤ 10)

###create list of AIC of the minimum of the AIC matrices, and find the
###minimum AIC of ARIMA models of 0≤ d ≤3
d.list<-list(d.0,d.1,d.2,d.3)
d.best<-which.min(d.list)

###find the row and column of the lowest AIC.
arrayInd(which.min(AIC.matrix[[d.best]]), dim(AIC.matrix[[d.best]]))
aic.col=arrayInd(which.min(AIC.matrix[[d.best]]), dim(AIC.matrix[[d.best]]))[2]
aic.row=arrayInd(which.min(AIC.matrix[[d.best]]), dim(AIC.matrix[[d.best]]))[1]

## store p,d,q of the best model
p.bestmodel=aic.row-1
d.bestmodel=d.best-1
q.bestmodel=aic.col-1



####fit the best model
best.model<-arima(x,order=c(p.bestmodel,d.bestmodel,q.bestmodel))

##plot the residuals of the best models acf and pacf.
acf(best.model$residuals)
pacf(best.model$residuals)
```

PART 2:

```
##install packages
install.packages("devtools")
install.packages("dplyr")
install.packages("pbapply")
install.packages("stringr")

devtools::install_github("hrbrmstr/omdbapi")

library(dplyr)
library(pbapply)
library(omdbapi)

getTv <- function(OUR_TITLE,OUR_YEAR=NA){
 x <- c()
 x.season <- c()
 x.episode <- c()

 #loop over seasons (Assuming maximum of 50 seasons)
 for(this.season in 1:50){
  #check if this season exists, otherwise break the for loop
  if(dim(find_by_title(OUR_TITLE,type="episode",
                season=this.season,
                episode=1,
                year_of_release = OUR_YEAR))[1] == 0){
   break
  } else {
   #now go over the episodes

   #first wait for 2 seconds (this amount probably needs to be higher)
   #(we don't want to get blacklisted from the API)
   print("Waiting for 2 seconds...")
   Sys.sleep(2)

   #looping over episodes (maximum is 50)
   for(this.episode in 1:50){
    if(dim(find_by_title(OUR_TITLE,
                type="episode",
                season=this.season,
                episode=this.episode,
                year_of_release = OUR_YEAR))[1] == 0){
     break
    } else {
     if(this.episode %% 9 ==0){
      print("Waiting for 2 seconds...")
      Sys.sleep(2)
     }
     this.rating <- find_by_title(OUR_TITLE,
                type="episode",
                season=this.season,
                episode=this.episode,
                year_of_release = OUR_YEAR)$imdbRating
     x <- c(x,this.rating)
     x.season <- c(x.season,this.season)
     x.episode <- c(x.episode,this.episode)
    }

   }

  }
 }

 return(data.frame(x=x,season=x.season,episode=x.episode))
}
```

```
##choose what tv show and name it OUR_TITLE
OUR_TITLE <-"The Wire"

##check if the show is on imdb
res.1<-search_by_title(OUR_TITLE,type="series")
res.1

#Check if it has ratings
find_by_title(OUR_TITLE, type="episode", season=1, episode=1)$imdbRating

Wire<-getTv(OUR_TITLE,OUR_YEAR = 2002)
x<-Wire$x


###FIT  a least squares model
lm1 <- lm(x~factor(season)-1,data=Wire)
##report the summary of the least squares model
summary(lm1)
## plot the residuals of the fitted model
x <- lm1$residuals

ts.plot(x)
acf(x)
pacf(x)


max.order <- 10
AIC.matrix <- list()


###Fit a series of ARIMA models with 0 ≤ p ≤ 10, 0 ≤ q ≤ 10, 1 ≤ d ≤ 3,
##and store the AIC of each fit.
max.d <- 3
for(d in 0:max.d){
 AIC.temp.matrix <- matrix(0,nrow = max.order+1,ncol= max.order+1)
 for(i in 1:(max.order+1)){
   for(j in 1:(max.order+1)){
    AIC.temp.matrix[i,j] <- tryCatch(
     {
      #try
      currentArima <- arima(x,order=c(i-1,d,j-1))
      AIC(currentArima)
     },
     error=function(cond){
      errMessage <- paste(i-1,d,j-1,sep=",")
      errMessage <- paste0("Error in fitting ARIMA(",errMessage,"), setting AIC to 10^6")
      message(errMessage)
      return(10^6)
     },
     warning=function(cond){
      errMessage <- paste(i-1,d,j-1,sep=",")
      errMessage <- paste0("Error in fitting ARIMA(",errMessage,"), setting AIC to 10^6")
      message(errMessage)
      return(10^6)
     })
   }
 }
 AIC.matrix[[d+1]] <- AIC.temp.matrix
}
```

```
###find the minimum aic of each AIC matrix, and store minimum aic of each matrix
d.0<-min(AIC.matrix[[1]]) ##ARIMA(0 ≤ p ≤ 10,0,0 ≤ q ≤ 10)
d.1<-min(AIC.matrix[[2]]) ##ARIMA(0 ≤ p ≤ 10,1,0 ≤ q ≤ 10)
d.2<-min(AIC.matrix[[3]]) ##ARIMA(0 ≤ p ≤ 10,2,0 ≤ q ≤ 10)
d.3<-min(AIC.matrix[[4]]) ##ARIMA(0 ≤ p ≤ 10,3,0 ≤ q ≤ 10)

###create list of AIC of the minimum of the AIC matrices, and find the
###minimum AIC of ARIMA models of 0≤ d ≤3
d.list<-list(d.0,d.1,d.2,d.3)
d.best<-which.min(d.list)


###find the 2nd and 3rd lowest AIC from the aic.matrix(d.best) with the lowest aic.
n <- length(AIC.matrix[[d.best]])
min.aic2<-sort(AIC.matrix[[d.best]],partial=n-119)[n-119]
min.aic3<-sort(AIC.matrix[[d.best]],partial=n-118)[n-118]

###find the row and column of the lowest AIC.
arrayInd(which.min(AIC.matrix[[d.best]]), dim(AIC.matrix[[d.best]]))
aic.col=arrayInd(which.min(AIC.matrix[[d.best]]), dim(AIC.matrix[[d.best]]))[2]
aic.row=arrayInd(which.min(AIC.matrix[[d.best]]), dim(AIC.matrix[[d.best]]))[1]

## store p,d,q of the best model
p.bestmodel=aic.row-1
d.bestmodel=d.best-1
q.bestmodel=aic.col-1


####fit the 3 best models of lowest aic
best.model<-arima(x,order=c(p.bestmodel,d.bestmodel,q.bestmodel))
best.model2<-arima(x,order=c(7,0,1))
best.model3<-arima(x,order=c(4,0,8))


##plot the residuals of the best model acf and pacf.
acf(best.model$residuals)
pacf(best.model$residuals)

##install stargazer which helps me create a table of the arima results to compare the 3 best models.
install.packages("stargazer")
library(stargazer)
regression_results<-stargazer(lm1,gls1, type="text",title = "Regression Results")
stargazer(best.model,best.model2,best.model3,type="text")
```

PART 3:

```
##install stargazer which helps me create a table of the arima results to compare the 3 best models.
install.packages("stargazer")
library(stargazer)

##install nlme to run gls regression
install.packages("nlme")
library(nlme)

###install vars package and load canada data set
install.packages("vars")
library(vars)
data(Canada)

##Save the columns of Canada as separate ts() variables, and change the names of the variables
C.e<-ts(Canada[,1],start=c(1980,1), end=c(2000,4), frequency=4 )# dependent variable,(employment)
C.prod<-ts(Canada[,2],start=c(1980,1), end=c(2000,4), frequency=4) #independent variable, (labour productivity)
C.rw<-ts(Canada[,3],start=c(1980,1), end=c(2000,4), frequency=4 ) #independent variable, (real wage)
C.U<-ts(Canada[,4],start=c(1980,1), end=c(2000,4), frequency=4 ) #independent variable, (unemplotment rate)

##difference the variables and store as new variables
diff.C.e<-diff(C.e)
diff.C.prod<-diff(C.prod)
diff.C.rw<-diff(C.rw)
diff.C.U<-diff(C.U)

###acf and pacf of all time series variables, including the differenced series
acf(C.e)
pacf(C.e)
acf(C.prod)
pacf(C.prod)
acf(C.rw)
pacf(C.rw)
acf(C.U)
pacf(C.U)
acf(diff.C.e)
pacf(diff.C.e)
acf(diff.C.prod)
pacf(diff.C.prod)
acf(diff.C.rw)
pacf(diff.C.rw)
acf(diff.C.U)
pacf(diff.C.U)

###Plot the ccf between C.e and all other covariates.
ccf(C.prod,C.e)##plot shows significant cross correlation in lag(0)
ccf(C.rw,C.e)##plot shows significant cross correlation in lag(0)
ccf(C.U,C.e)##plot shows significant cross correlation at lag(0),lag(-4),lag(1),lag(2)
ccf(diff.C.prod,C.e)##plot shows no significant cross correlation
ccf(diff.C.rw,C.e)###plot shows significant cross correlation at lag(-2),lag(-1)
ccf(diff.C.U,C.e)###shows significant cross correlation at lag(-1)

###create lag versions of variables that have signifcant cross correlation.
C.U.lag.1<-lag(C.U[-84],-4)
C.U.lag.2<-lag(C.U[-84],1)
C.U.lag.3<-lag(C.U[-84],2)
diff.C.rw.lag.1<-lag(diff.C.rw,-2)
diff.C.rw.lag.2<-lag(diff.C.rw,-1)
diff.C.U.lag1<-lag(diff.C.U,-1)

###fit a least squares regression model with the covariates that have a significant cross correlation with C.e
###make sure all the variable are thame same length
C.e<-C.e[-84]
C.U<-C.U[-84]
C.prod<-C.prod[-84]
```

```
C.rw<-C.rw[-84]

lm.1<-lm(C.e~C.prod+C.rw+C.U+C.U.lag.1+C.U.lag.2+C.U.lag.3+diff.C.rw.lag.1
       + diff.C.rw.lag.2+diff.C.U.lag1)

##Plot the acf and the pacf of the residuals of the fitted least squares model
x<-lm.1$residuals
acf(x)
pacf(x)

###Fit a series of ARIMA models with 0 ≤ p ≤ 10, 0 ≤ q ≤ 10, 1 ≤ d ≤ 3,
##and store the AIC of each fit.
max.order <- 10
AIC.matrix <- list()
max.d <- 3
for(d in 0:max.d){
 AIC.temp.matrix <- matrix(0,nrow = max.order+1,ncol= max.order+1)
 for(i in 1:(max.order+1)){
  for(j in 1:(max.order+1)){
   AIC.temp.matrix[i,j] <- tryCatch(
    {
     #try
     currentArima <- arima(x,order=c(i-1,d,j-1))
     AIC(currentArima)
    },
    error=function(cond){
     errMessage <- paste(i-1,d,j-1,sep=",")
     errMessage <- paste0("Error in fitting ARIMA(",errMessage,"), setting AIC to 10^6")
     message(errMessage)
     return(10^6)
    },
    warning=function(cond){
     errMessage <- paste(i-1,d,j-1,sep=",")
     errMessage <- paste0("Error in fitting ARIMA(",errMessage,"), setting AIC to 10^6")
     message(errMessage)
     return(10^6)
    })
  }
 }
 AIC.matrix[[d+1]] <- AIC.temp.matrix
}


###find the minimum aic of each AIC matrix, and store minimum aic of each matrix
d.0<-min(AIC.matrix[[1]]) ##ARIMA(0 ≤ p ≤ 10,0,0 ≤ q ≤ 10)
d.1<-min(AIC.matrix[[2]]) ##ARIMA(0 ≤ p ≤ 10,1,0 ≤ q ≤ 10)
d.2<-min(AIC.matrix[[3]]) ##ARIMA(0 ≤ p ≤ 10,2,0 ≤ q ≤ 10)
d.3<-min(AIC.matrix[[4]]) ##ARIMA(0 ≤ p ≤ 10,3,0 ≤ q ≤ 10)


###create list of AIC of the minimum of the AIC matrices, and find the
###minimum AIC of ARIMA models of 0≤ d ≤3
d.list<-list(d.0,d.1,d.2,d.3)
d.best<-which.min(d.list)

###find the row and column of the lowest AIC.
arrayInd(which.min(AIC.matrix[[d.best]]), dim(AIC.matrix[[d.best]]))
aic.col=arrayInd(which.min(AIC.matrix[[d.best]]), dim(AIC.matrix[[d.best]]))[2]
aic.row=arrayInd(which.min(AIC.matrix[[d.best]]), dim(AIC.matrix[[d.best]]))[1]

## store p,d,q of the best model
p.bestmodel=aic.row-1
d.bestmodel=d.best-1
q.bestmodel=aic.col-1

###fit a arima model, with the parameters of the best model with the lowest aic
```

```
best.model <- arima(C.e,order=c(p.bestmodel,d.bestmodel,q.bestmodel))

##fit a generalized least squares model using the order of the best model we just fitted
gls1 <- gls(C.e~C.prod + C.rw + C.U + C.U.lag.1 + C.U.lag.2 + C.U.lag.3
        + diff.C.rw.lag.1
        + diff.C.rw.lag.2 + diff.C.U.lag1,
        data=Canada,
        correlation = corARMA(value = .5*coef(best.model)[1:6],p=2,q=4),
        control = list(singular.ok = TRUE) )

##look at the summary of the generalized least square and least squares model and comparethem.
summary(gls1)
####create table of results of gls model
regression_results<-stargazer(gls1, type="text",title = "Regression Results")
```