# ECO374 PS1

Shih-Chieh Lee

2023-03-08

## Import all required packages

## 1. Read FRED Data in Class XTS

Data: https://fred.stlouisfed.org/series/TLNRESCONS, adjusted to 2005-01-01 and the end to 2022-12-01

```
start = as.Date("2005-01-01")
end = as.Date("2022-12-01")
table <- read.csv(file = "TLNRESCONS.csv", header = TRUE, sep = ",")
time.chr <- paste(table$DATE)
time.dt <- as.Date(time.chr, format="%Y-%m-%d")
Construction <- xts(x = table$TLNRESCONS, order.by = time.dt)
class(Construction)
```

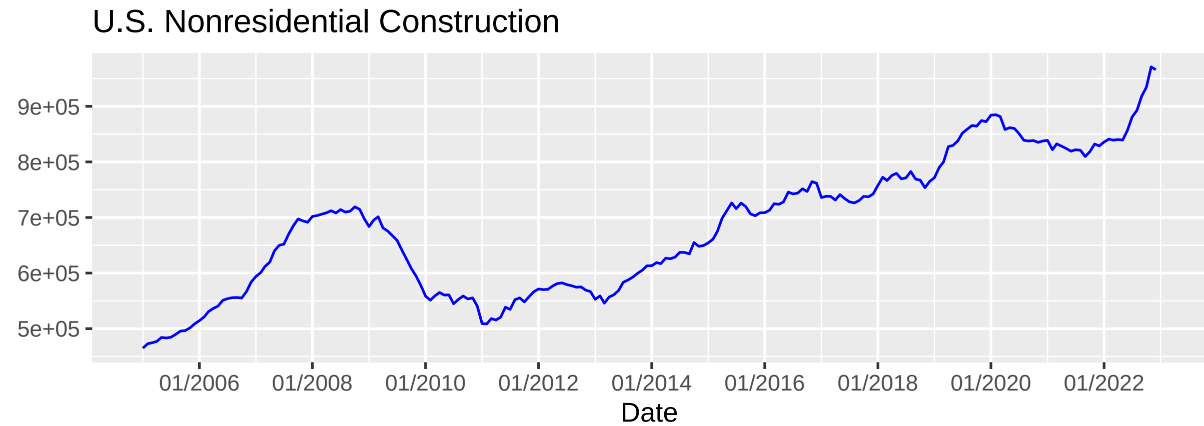
```
## [1] "xts" "zoo"
```

The data is now loaded in xts. class

## 2. & 3. Plot Data

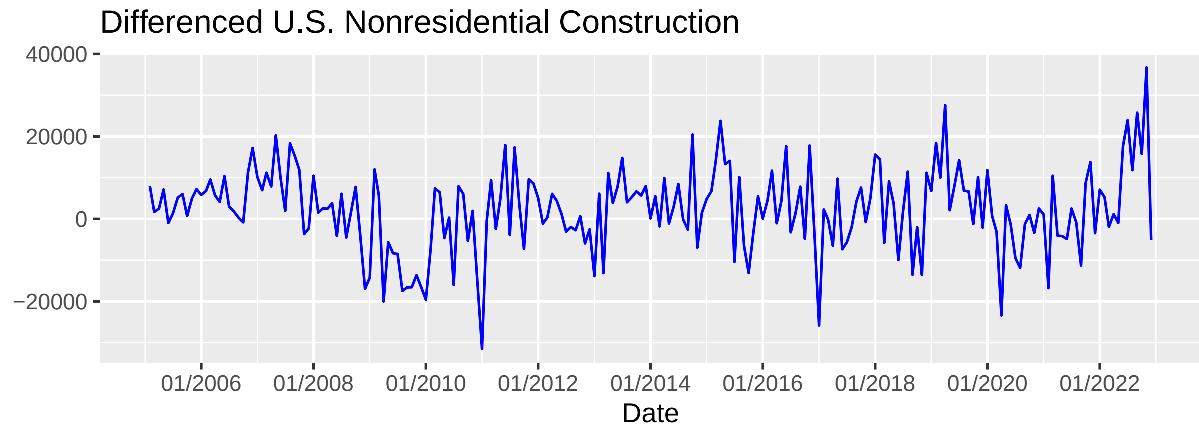
We start from plotting the original data:

```
ggplot(data = Construction, aes(x=index(Construction), y = Construction)) +
  geom_line(color = "blue", linetype = "solid") +
  labs(x = "Date" , y = "", title = "U.S. Nonresidential Construction") +
  scale_x_date(date_breaks = "2 years", date_labels = "%m/%Y")
```




Next, difference the data and plot again:

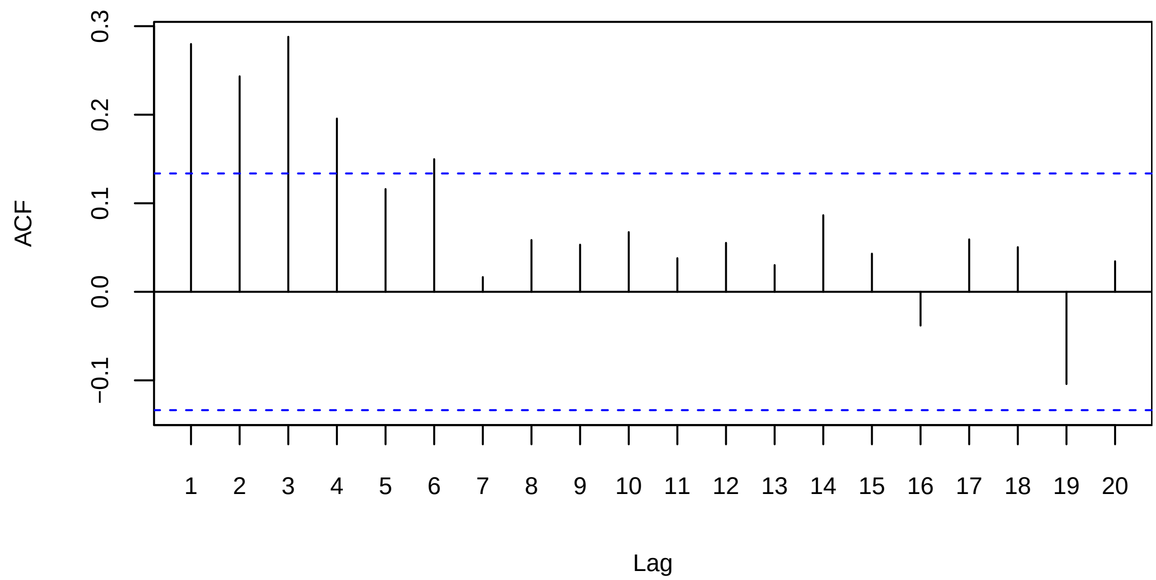
```
Con_diff <- diff(Construction)
Con_diff <- na.omit(Con_diff)
ggplot(data = Con_diff, aes(x=index(Con_diff), y = Con_diff)) +
  geom_line(color = "blue", linetype = "solid") +
  labs(x = "Date" , y = "", title = "Differenced U.S. Nonresidential Construction") +
  scale_x_date(date_breaks = "2 years", date_labels = "%m/%Y")
```



## 4. ACF & PACF:

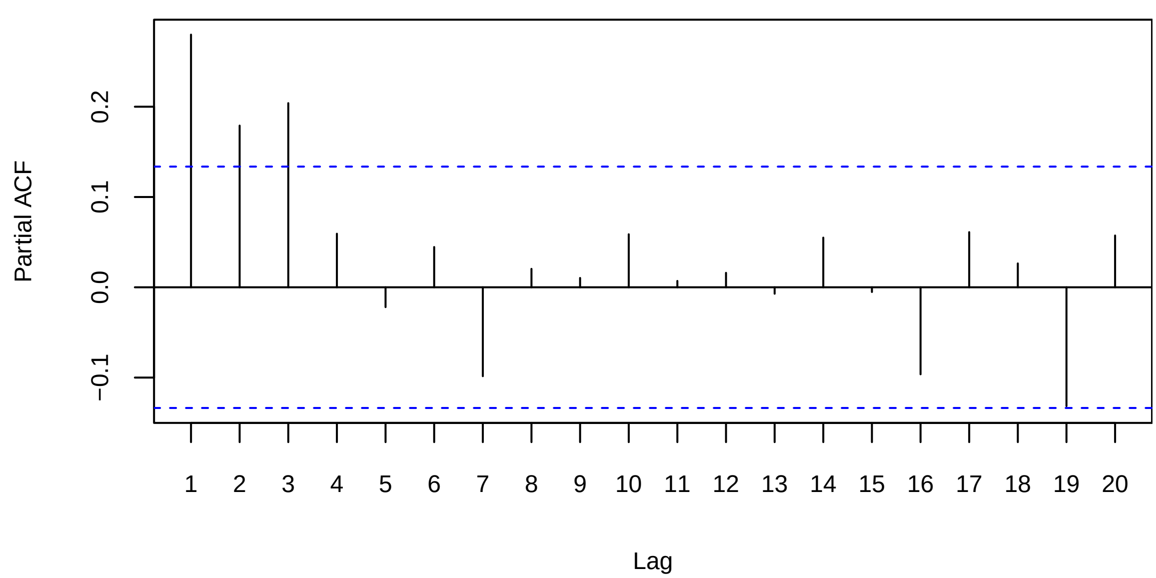
We now move to plot the ACF and PACF of the differenced data:

```
par(mar=c(4,4,0.5,0))
ACF <- acf(Con_diff, lag.max=20, plot=FALSE, demean=TRUE)
plot(ACF[1:20], main="", cex.lab=0.75, cex.axis=0.75, xaxt="n")
axis(1,at=ACF$lag, cex.axis=0.75)
```




Then, we plot the PACF of the data

```
par(mar=c(4,4,0.5,0))
PACF <- pacf(Con_diff, lag.max=20, plot=FALSE, demean=TRUE)
plot(PACF[1:20], main="", cex.lab=0.75, cex.axis=0.75, xaxt="n")
axis(1,at=PACF$lag, cex.axis=0.75)
```



# 5-7. Model Selection:

Q5: Best-fitting ARMA model on the differenced data set

```
auto.arima(Con_diff)
```

```
## Series: Con_diff
## ARIMA(1,0,1) with zero mean
##
## Coefficients:
##          ar1      ma1
##       0.8951  -0.6818
## s.e.  0.0577   0.0892
##
## sigma^2 = 86258078:  log likelihood = -2268.58
## AIC=4543.15   AICc=4543.27   BIC=4553.27
```

Q6: Best-fitting NNAR model on the differenced data set + NNAR cross-validation

We first introduce the NNAR model to the data

```
NNAR.model <- nnetar(y = Con_diff)
NNAR.model
```

```
## Series: Con_diff
## Model:  NNAR(3,2)
## Call:   nnetar(y = Con_diff)
##
## Average of 20 networks, each of which is
## a 3-2-1 network with 11 weights
## options were - linear output units
##
## sigma^2 estimated as 74439510
```

Then perform time-series cross-validation

```
TSCV_nnetar <- function(Con_diff, p, P, size) {
  TT <- length(Con_diff)
  T1 <- floor(TT/5) # start at 20% of the sample size
  step <- 20 # forecast horizon for MSE
  MSE.t <- matrix(0,nrow=TT-T1+1,ncol=1) # initialize
  y.hat <- MSE.t # initialize
  tseq <- seq(from=T1, to=TT, by=step)
  tseq <- tseq[-length(tseq)]
  for (j in tseq) {
    #print(j) # display progress through data
    set.seed(seed)
    nnetar.model <- nnetar(y=data[1:j-1], p=3, P=2, size=size) # fit nnetar model on the training set
    NN.f <- forecast(nnetar.model,h=step) # generate forecast
    y.hat <- as.numeric(NN.f$mean)
    js <- j+step-1
    MSE.t[(j-T1+1):(js-T1+1)] <- (as.numeric(data[j:js])-y.hat)^2
```

```
  }
  MSE.validation <- mean(MSE.t)
  return(MSE.validation)
}
```

Time-series validation MSE for Specified Models

```
data <- Con_diff

# Loop over different model specifications
for (m in 1:6) {

  TT <- length(data)
  T1 <- floor(0.2*TT) # start at 20% of the sample size
  step <- 20 # forecast data horizon for MSE
  MSE.t <- matrix(0,nrow=TT-T1+1,ncol=1) # initialize
  MAE.t <- MSE.t
  MAPE.t<- MSE.t
  tseq <- seq(from=T1, to=TT, by=step)
  tseq <- tseq[-length(tseq)]

  for (j in tseq) {

    # auto.arima model: ARMA(1,1)
    if (m==1) {fcst <- forecast(arima(data[1:j-1], order=c(1,0,1)), h=step)
               yhat <- as.numeric(fcst[[4]])}

    # ARMA(2,0)
    if (m==2) {fcst <- forecast(arima(data[1:j-1], order=c(2,1,0)), h=step)
              # the fcst$mean forecast is stored in the 4th element of the list fcst
               yhat <- as.numeric(fcst[[4]])}

    # S-ARMA(2,0) with one seasonal AR component at frequency of 6 (semi-annunal)
    if (m==3) {fcst <- forecast(arima(data[1:j-1], order=c(2,1,0), seasonal = list(order = c(1,0,0), pe
               yhat <- as.numeric(fcst[[4]])}

    # SETAR model with a threshold of 0
    if (m==4) {fcst <- predict(setar(data[1:j-1], mL=1, mH=1, th=0), n.ahead=step)
               yhat <- as.numeric(fcst)
               yhat <- cumsum(yhat) + as.numeric(last(data[1:j-1]))} # cumulate forecast differences

    # LSTAR model with all parameters set to 1
    if (m==5) {fcst <- predict(lstar(data[1:j-1], m=1, d=1, mL=1, mH=1, gamma=1, th=1, trace=FALSE), n.a
               yhat <- as.numeric(fcst)
               yhat <- cumsum(yhat) + as.numeric(last(data[1:j-1]))} # cumulate forecast differences

    # NNAR with parameters selected previously
    if (m==6) {fcst <- forecast(nnetar(data[1:j-1], 3, 2, 5), h=step)
               # the fcst$mean forecast is stored in the 16th element of the list fcst
               yhat <- as.numeric(fcst[[16]][1:step])}

    js <- j+step-1
    MSE.t[(j-T1+1):(js-T1+1)] <- (as.numeric(data[j:js])-yhat)^2
```

```
    MAE.t[(j-T1+1):(js-T1+1)] <- abs(as.numeric(data[j:js])-yhat)
    MAPE.t[(j-T1+1):(js-T1+1)] <- 100*abs((as.numeric(data[j:js])-yhat)/yhat)
  }

  if (m<=3) print(fcst$method)
  if (m==4) print("SETAR")
  if (m==5) print("LSTAR")
  if (m==6) print("NNAR")

  print(paste("MSE  =", mean(MSE.t)))
  print(paste("MAE  =", mean(MAE.t)))
  print(paste("MAPE =", mean(MAPE.t)))
  print(" ")
}
```

```
## [1] "ARIMA(1,0,1) with non-zero mean"
## [1] "MSE  = 105765712.35975"
## [1] "MAE  = 7498.38452297429"
## [1] "MAPE = 296.442520211481"
## [1] " "
## [1] "ARIMA(2,1,0)"
## [1] "MSE  = 116192466.913854"
## [1] "MAE  = 8306.36286535745"
## [1] "MAPE = 343.041586733881"
## [1] " "
## [1] "ARIMA(2,1,0)(1,0,0)[6]"
## [1] "MSE  = 121220643.210978"
## [1] "MAE  = 8537.69719144872"
## [1] "MAPE = 5381.68765673848"
## [1] " "
## [1] "SETAR"
## [1] "MSE  = 1880574093.43441"
## [1] "MAE  = 35325.4116223698"
## [1] "MAPE = 98.869773609081"
## [1] " "
## [1] "LSTAR"
## [1] "MSE  = 1880574093.43441"
## [1] "MAE  = 35325.4116223698"
## [1] "MAPE = 98.8697736090811"
## [1] " "
## [1] "NNAR"
## [1] "MSE  = 122830971.001836"
## [1] "MAE  = 8239.66814506705"
## [1] "MAPE = 286.36350571084"
## [1] " "
```

# 8. Comment

From the results above, we are able to determine that the first model (auto.arima best-fitting lienar combination model) has the lowest MSE as well as the lowest MAE. However, it is the LSTAR and SETAR models who share the lowest level of MAPE.