Graded Problem Set 1

November 2nd, 2023

```r
if (!require("quantmod")) install.packages("quantmod")
```

```
## Loading required package: quantmod

## Loading required package: xts

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: TTR

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
if (!require("stats")) install.packages("stats")
if (!require("forecast")) install.packages("forecast")  # Neural Networks
```

```
## Loading required package: forecast
```

```r
if (!require("timetk")) install.packages("timetk")
```

```
## Loading required package: timetk
```

```r
if (!require("ggplot2")) install.packages("ggplot2")
```

```
## Loading required package: ggplot2
```

```r
if (!require("tsDyn")) install.packages("tsDyn")
```

```
## Loading required package: tsDyn
```

```r
library(quantmod) # functions: getSymbols
library(ggplot2)  # functions: ggplot
library(stats)    # functions: arima
library(tsDyn)    # functions: SETAR
library(forecast) # functions: auto.arima, nnetar
library(urca)     # functions: ur.kpss
library(timetk)
```

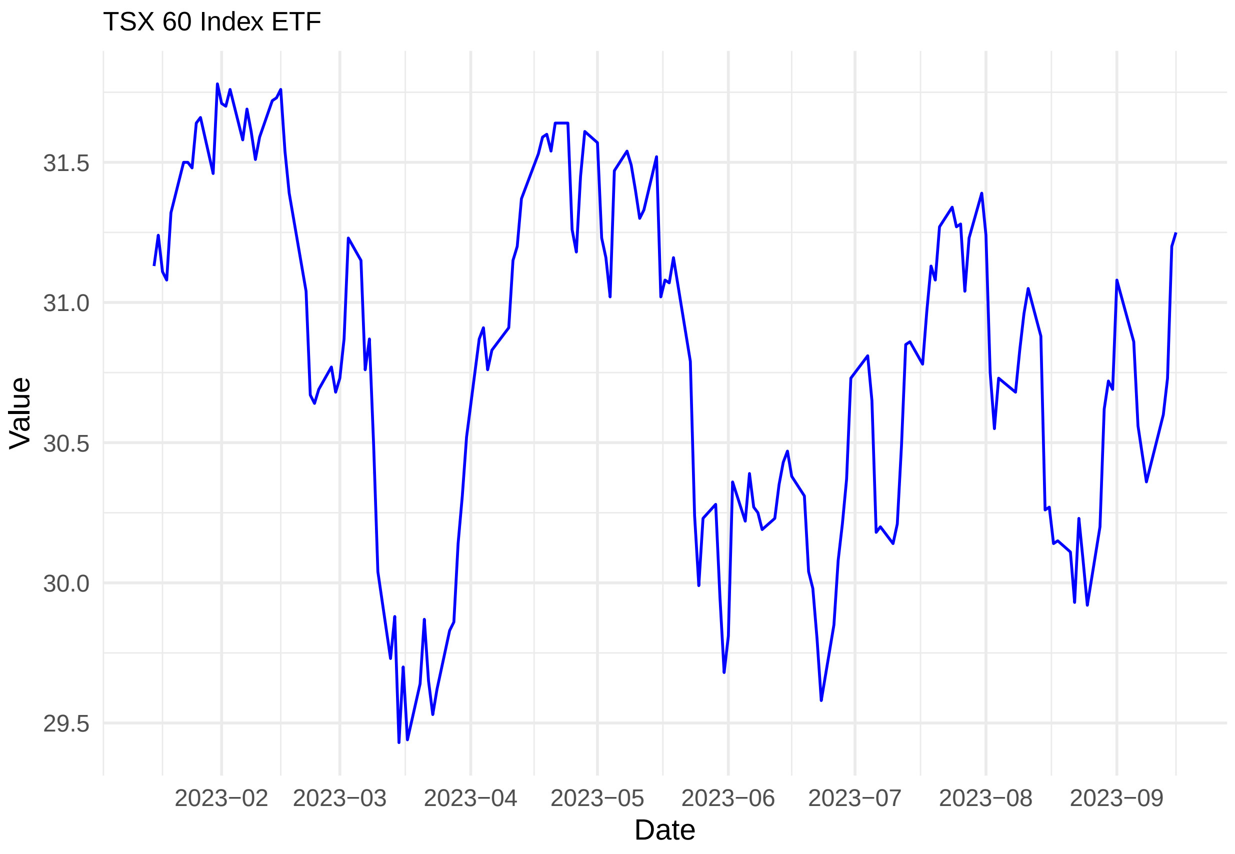
1. Data

(Source)

```r
XIU <- getSymbols("XIU.TO", src="yahoo", return.class="xts", auto.assign=F)
XIU <- window(XIU, start=as.Date("2023-1-15"), end=as.Date("2023-09-15"))
XIU.c <- XIU$XIU.TO.Close # extract close price
seed <- 2345
```

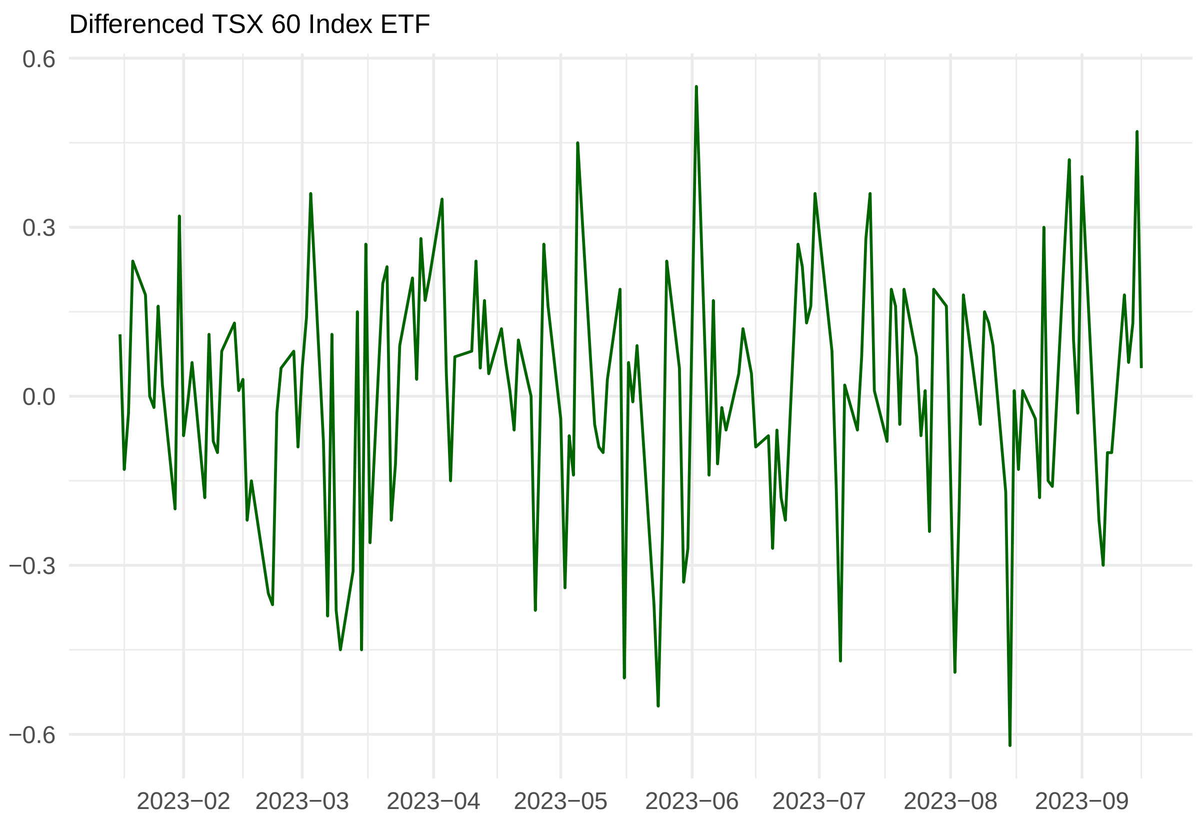
Plot Data

```r
ggplot(XIU.c, aes(x=index(XIU.c), y=XIU.TO.Close)) + geom_line(color="blue") +
  labs(x="Date", y="Value", title="TSX 60 Index ETF") +
  theme_minimal() + theme(plot.title = element_text(size=10)) +
  scale_x_date(date_breaks="1 months", date_labels = "%Y-%m")
```



Difference Data

```r
D_XIU.c <- na.omit(diff(XIU.c, lag=1, differences=1))
ggplot(D_XIU.c, aes(x=index(D_XIU.c), y=XIU.TO.Close)) + geom_line(color="darkgreen") +
  labs(x="", y="", title="Differenced TSX 60 Index ETF") +
  theme_minimal() + theme(plot.title = element_text(size=10)) +
  scale_x_date(date_breaks="1 months", date_labels = "%Y-%m")
```



Unit Root Test

```r
ur.test <- ur.kpss(XIU.c)
summary(ur.test)
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 4 lags.
##
## Value of test-statistic is: 0.4671
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```r
ur.test.d <- ur.kpss(D_XIU.c)
summary(ur.test.d)
```

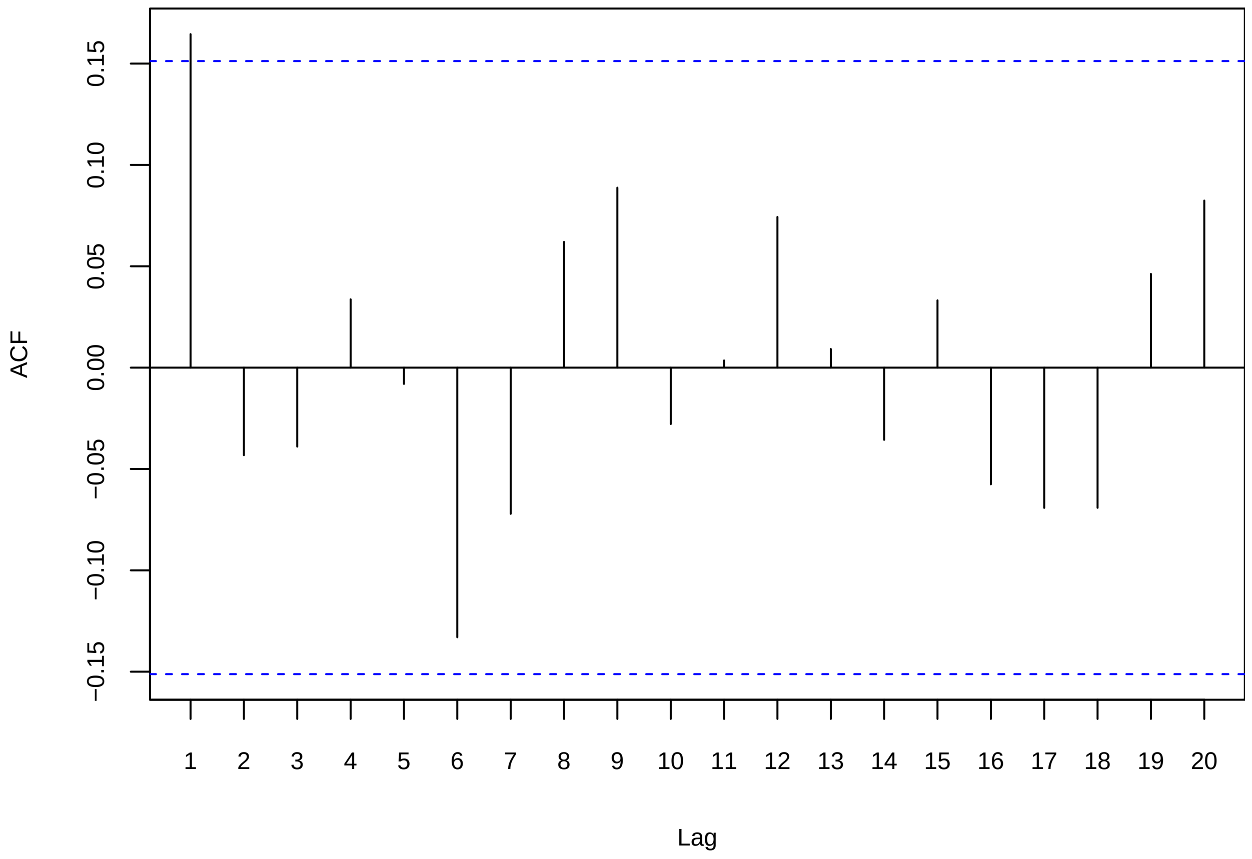
```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 4 lags.
##
## Value of test-statistic is: 0.0629
##
```

```
## Critical value for a significance level of:
##                  10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

2. PACF and ACF

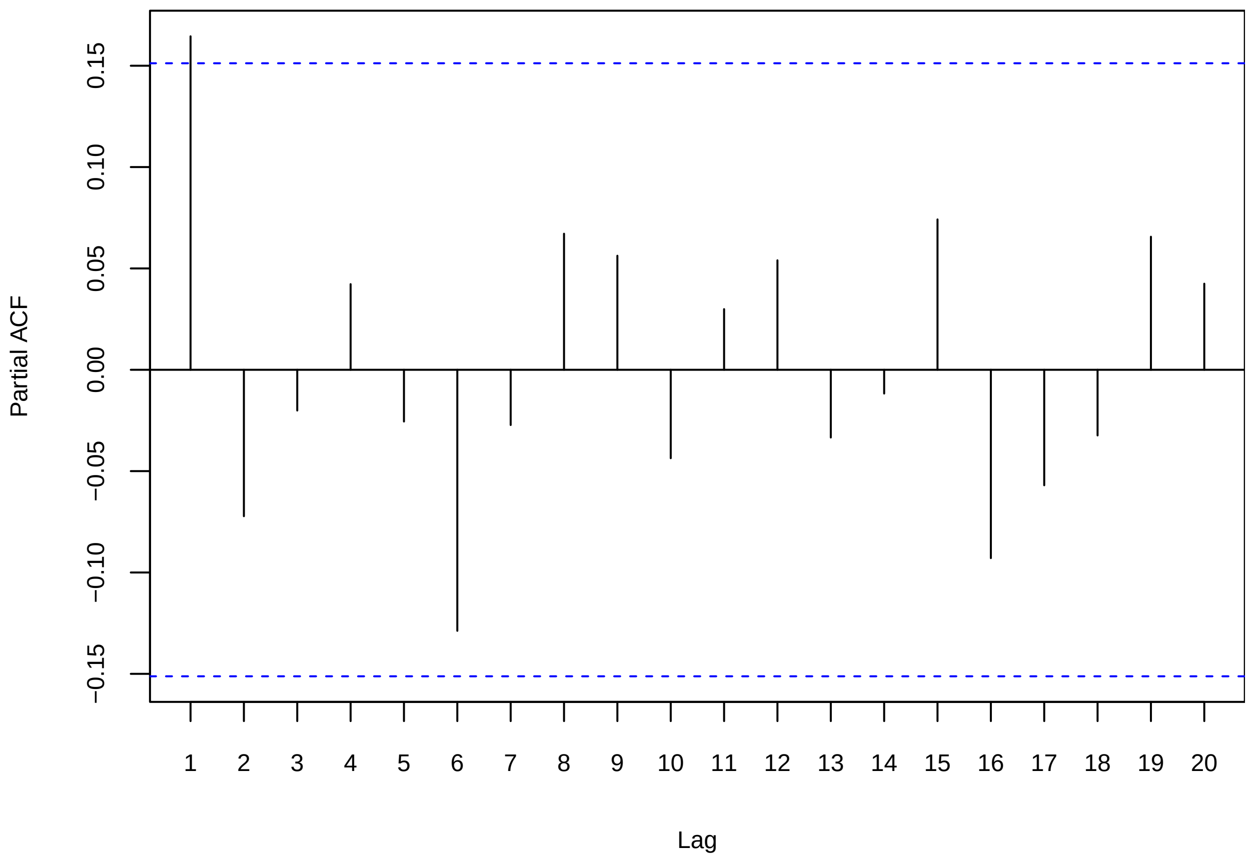
ACF

```
par(mar=c(4,4,0.5,0)) # set margin sizes
ACF <- acf(D_XIU.c, 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) # put a label at each lag value
```



PACF

```
par(mar=c(4,4,0.5,0)) # set margin sizes
PACF <- pacf(D_XIU.c, 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) # put a label at each lag value
```



3 Model Selection

ARIMA on Full Data Set

```
auto.arima(XIU.c)
```

```
## Series: XIU.c
## ARIMA(0,1,1)
##
## Coefficients:
##          ma1
##       0.1806
## s.e.  0.0772
##
## sigma^2 = 0.04407:  log likelihood = 24.36
## AIC=-44.72   AICc=-44.64   BIC=-38.47
```

ARIMA on Differenced Data

```
auto.arima(D_XIU.c)
```

```
## Series: D_XIU.c
## ARIMA(0,0,1) with zero mean
##
## Coefficients:
##          ma1
##       0.1806
## s.e.  0.0772
```

```
##
## sigma^2 = 0.04407:  log likelihood = 24.36
## AIC=-44.72   AICc=-44.64   BIC=-38.47
```

Fit NNAR model

```r
seed <- 2345
set.seed(seed)
nnetar.model <- nnetar(y=XIU.c) # fit nnetar model on the training set
nnetar.model
```

```
## Series: XIU.c
## Model:  NNAR(2,2)
## Call:   nnetar(y = XIU.c)
##
## Average of 20 networks, each of which is
## a 2-2-1 network with 9 weights
## options were - linear output units
##
## sigma^2 estimated as 0.03942
```

```r
y.hat <- forecast(nnetar.model, h=1) # generate forecast 1 time step ahead
y.hat$mean
```

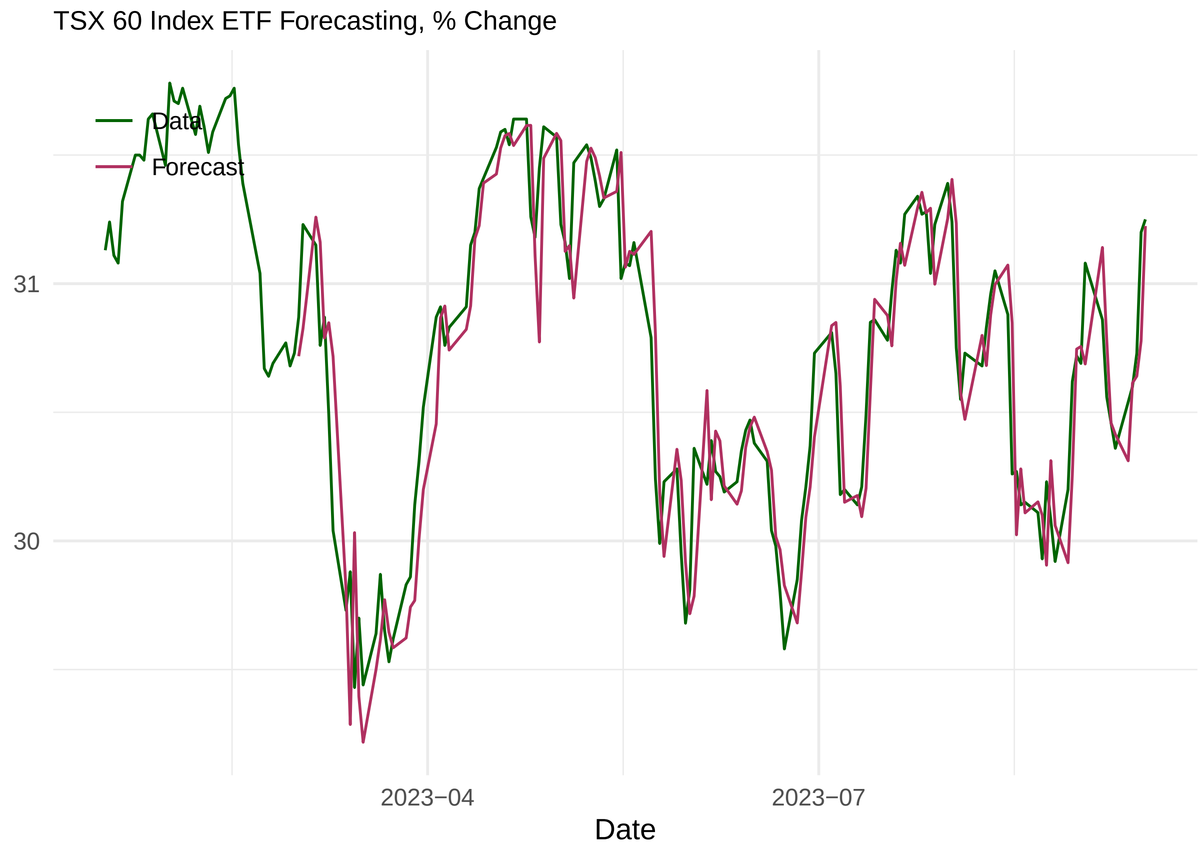
```
## Time Series:
## Start = 170
## End = 170
## Frequency = 1
## XIU.TO.Close
##     31.23711
```

```r
data <- XIU.c
TT <- length(data)
T1 <- floor(0.2*TT) # start at 20% of the sample size
y.hat <- matrix(0,nrow=TT-T1+1,ncol=1) # initialize

for (j in T1:TT) {
  #print(j) # display progress through data
  nnetar.model <- nnetar(y=data[1:j-1]) # fit nnetar model on the training set
  NN.f <- forecast(nnetar.model,h=1) # generate forecast 1 time step ahead
  y.hat[j-T1+1] <- as.numeric(NN.f$mean)
}

XIU.pc.s <- XIU.c
XIU.pc.f <- xts(y.hat, order.by=index(XIU.pc.s[T1:TT]))
colnames(XIU.pc.f) <- "forecast"

colors <- c("Data"="darkgreen", "Forecast"="maroon") # set colors for plotted lines
ggplot() +
  geom_line(data=XIU.pc.s, aes(x=index(XIU.pc.s), y=XIU.TO.Close, color="Data")) +
  geom_line(data=XIU.pc.f, aes(x=index(XIU.pc.f), y=forecast, color="Forecast")) +
  labs(x="Date", y="", title="TSX 60 Index ETF Forecasting, % Change") +
  theme_minimal() +
  theme(plot.title = element_text(size=10)) +
  theme(legend.position = c(0.1, 0.9)) +
  scale_color_manual(name="", values=colors)  +
  scale_x_date(date_breaks="3 months", date_labels = "%Y-%m")
```



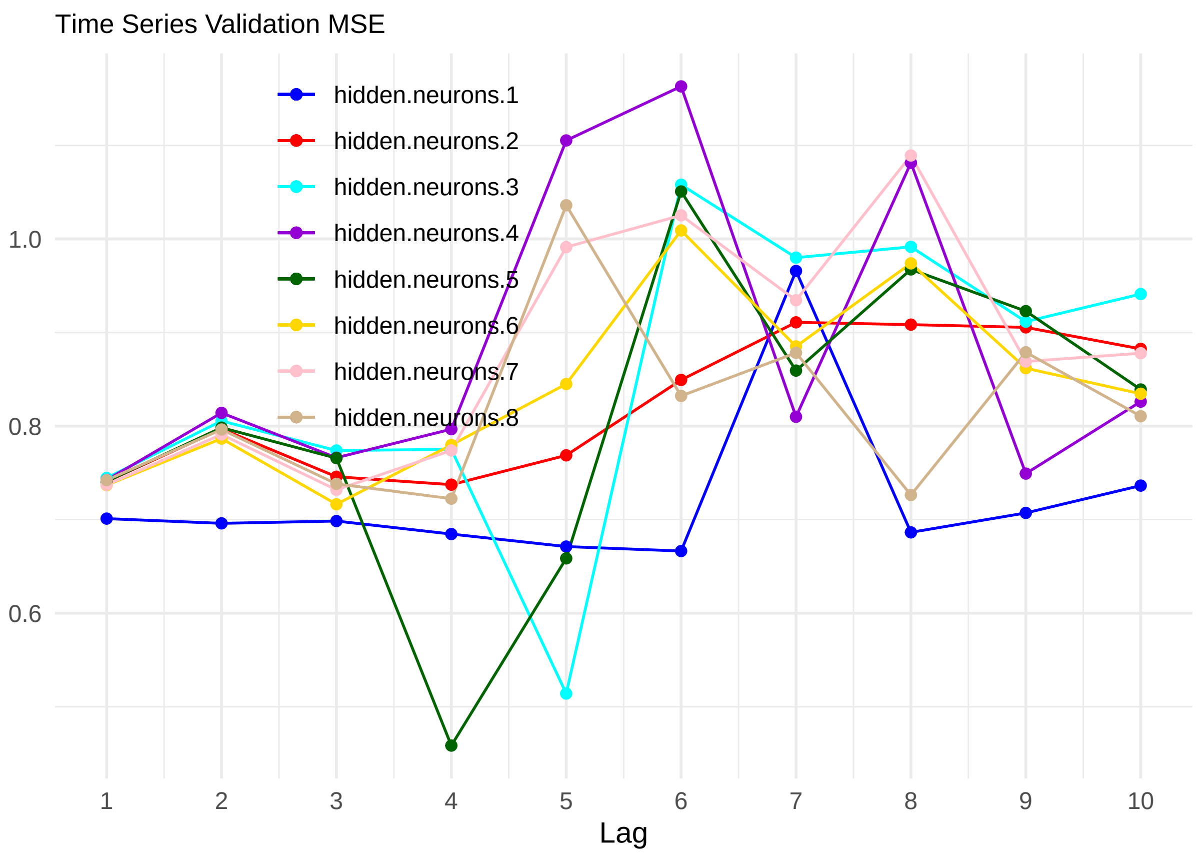
```
TSCV_nnetar <- function(data, p, P, size) {
  TT <- length(data)
  T1 <- floor(TT/5) # start at 20% of the sample size
  step <- 20 # forecast horizon for MSE
  MSE.t <- matrix(0,nrow=TT-T1+1,ncol=1) # initialize
  y.hat <- MSE.t # initialize
  tseq <- seq(from=T1, to=TT, by=step)
  tseq <- tseq[-length(tseq)]
  for (j in tseq) {
    #print(j) # display progress through data
    set.seed(seed)
    nnetar.model <- nnetar(y=data[1:j-1], p=p, P=P, 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)
}
```

```
size.max <- 8
lag.max <- 10
TSCV <- matrix(0, nrow=lag.max, ncol=size.max)
for (s in 1:size.max) {
  for (k in 1:lag.max) {
```

```r
    TSCV[k,s] <- TSCV_nnetar(data=XIU.c, p=k, P=0, size=s)
    #writeLines(paste("Size = ", s, "Lag = ", k, "  Validation MSE = ", TSCV[k,s]))
  }}
TSCV <- data.frame(TSCV)
colnames(TSCV) <- paste("hn.", 1:size.max , sep="")
```

Time Series Validation NNAR

```r
colors <- c("hidden.neurons.1"="blue", "hidden.neurons.2"="red",
            "hidden.neurons.3"="cyan", "hidden.neurons.4"="darkviolet",
            "hidden.neurons.5"="darkgreen", "hidden.neurons.6"="gold",
            "hidden.neurons.7"="pink", "hidden.neurons.8"="tan")
ggplot() +
  geom_line(data=TSCV, aes(x=index(TSCV), y=hn.1, color="hidden.neurons.1")) +
  geom_line(data=TSCV, aes(x=index(TSCV), y=hn.2, color="hidden.neurons.2")) +
  geom_line(data=TSCV, aes(x=index(TSCV), y=hn.3, color="hidden.neurons.3")) +
  geom_line(data=TSCV, aes(x=index(TSCV), y=hn.4, color="hidden.neurons.4")) +
  geom_line(data=TSCV, aes(x=index(TSCV), y=hn.5, color="hidden.neurons.5")) +
  geom_line(data=TSCV, aes(x=index(TSCV), y=hn.6, color="hidden.neurons.6")) +
  geom_line(data=TSCV, aes(x=index(TSCV), y=hn.7, color="hidden.neurons.7")) +
  geom_line(data=TSCV, aes(x=index(TSCV), y=hn.8, color="hidden.neurons.8")) +
  geom_point(data=TSCV, aes(x=index(TSCV), y=hn.1, color="hidden.neurons.1")) +
  geom_point(data=TSCV, aes(x=index(TSCV), y=hn.2, color="hidden.neurons.2")) +
  geom_point(data=TSCV, aes(x=index(TSCV), y=hn.3, color="hidden.neurons.3")) +
  geom_point(data=TSCV, aes(x=index(TSCV), y=hn.4, color="hidden.neurons.4")) +
  geom_point(data=TSCV, aes(x=index(TSCV), y=hn.5, color="hidden.neurons.5")) +
  geom_point(data=TSCV, aes(x=index(TSCV), y=hn.6, color="hidden.neurons.6")) +
  geom_point(data=TSCV, aes(x=index(TSCV), y=hn.7, color="hidden.neurons.7")) +
  geom_point(data=TSCV, aes(x=index(TSCV), y=hn.8, color="hidden.neurons.8")) +
  labs(x="Lag", y="", title="Time Series Validation MSE") +
  theme_minimal() +
  theme(plot.title = element_text(size=10)) +
  theme(legend.position = c(0.3, 0.75)) +
  scale_color_manual(name="", values=colors) +
  scale_x_continuous(breaks=index(TSCV), labels=paste(index(TSCV)))
```



```
data <- XIU.c
D_data <- D_XIU.c

# 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) {

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

    if (m==2) {fcst <- forecast(arima(data[1:j-1], order=c(1,1,1)), h=step)

               yhat <- as.numeric(fcst[[4]])}

    if (m==3) {fcst <- forecast(arima(data[1:j-1], order=c(1,1,1), seasonal=c(1,1,0)), h=step)
               yhat <- as.numeric(fcst[[4]])}
```

```r
    if (m==4) {fcst <- predict(setar(D_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]))}
    if (m==5) {fcst <- predict(lstar(D_data[1:j-1], m=1, d=1, mL=1, mH=1,th=0,gamma=1, trace=FALSE), n.a
               yhat <- as.numeric(fcst)
               yhat <- cumsum(yhat) + as.numeric(last(data[1:j-1]))}

    if (m==6) {set.seed(seed)
               fcst <- forecast(nnetar(data[1:j-1], p=4, P=0, size=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(0,1,1)"
## [1] "MSE  = 0.621530067379266"
## [1] "MAE  = 0.595157294665062"
## [1] "MAPE = 1.95722578844188"
## [1] " "
## [1] "ARIMA(1,1,1)"
## [1] "MSE  = 0.603159781065101"
## [1] "MAE  = 0.587178487045255"
## [1] "MAPE = 1.92948231822366"
## [1] " "
## [1] "ARIMA(1,1,1)"
## [1] "MSE  = 0.759647260508624"
## [1] "MAE  = 0.643270583078029"
## [1] "MAPE = 2.12102568591447"
## [1] " "
## [1] "SETAR"
## [1] "MSE  = 0.624050058035389"
## [1] "MAE  = 0.60586870778566"
## [1] "MAPE = 1.98069807866385"
## [1] " "
## [1] "LSTAR"
## [1] "MSE  = 0.673623064942327"
## [1] "MAE  = 0.629301185843958"
## [1] "MAPE = 2.05941989194158"
## [1] " "
## [1] "NNAR"
## [1] "MSE  = 0.458513717844784"
## [1] "MAE  = 0.515427675277786"
## [1] "MAPE = 1.67968309668533"
## [1] " "
```

The NNAR(4,5) model yields the lowest MSE, MAE, and MAPE. The model uses 5 hidden layers and 4 lags for optimal forecasting. The MSE is 0.458, MAE is 0.515, and MAPE is 1.67.