

Quant

David Bowden

2024-03-30

```
library(dplyr)
```

```
##  
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':  
##  
##   filter, lag
```

```
## The following objects are masked from 'package:base':  
##  
##   intersect, setdiff, setequal, union
```

```
df <- read.csv("C:\\Users\\david\\Downloads\\DAL_data_1.csv")  
df2 <- read.csv("C:\\Users\\david\\Downloads\\CLF_data_1.csv")  
CLF <- df %>%  
  semi_join(df2, by = "Date")  
  
DAL <- df2 %>%  
  semi_join(df, by = "Date")  
  
length(CLF$Close)
```

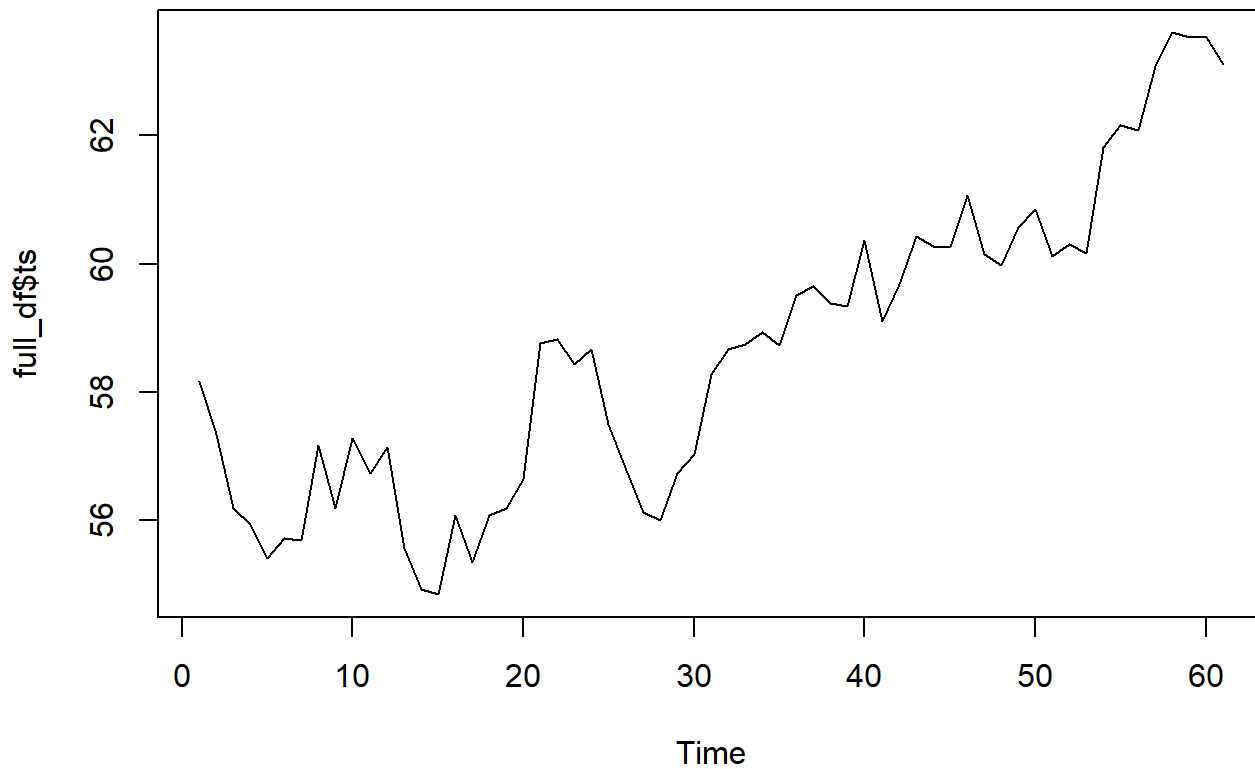
```
## [1] 2516
```

```
length(DAL$Close)
```

```
## [1] 2516
```

```
full_df <- data.frame(DAL, CLF)  
full_df$combined <- c((DAL$Close + CLF$Close) / 2)  
full_df$relative_strength <- ((DAL$Close + CLF$Close)/2 - (DAL$Low + CLF$Low)/2) / ((DAL$High + CLF$High) / 2 - (DAL$Low + CLF$Low)/2)  
full_df <- full_df[(length(full_df$Close) - 60):length(full_df$Close),]  
ts <- ts(full_df$combined)  
time <- time(ts)  
full_df$ts <- ts  
full_df$time <- time  
plot(full_df$ts, main="Time Series")
```

Time Series



```
library(MASS)
```

```
##  
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':  
##  
##   select
```

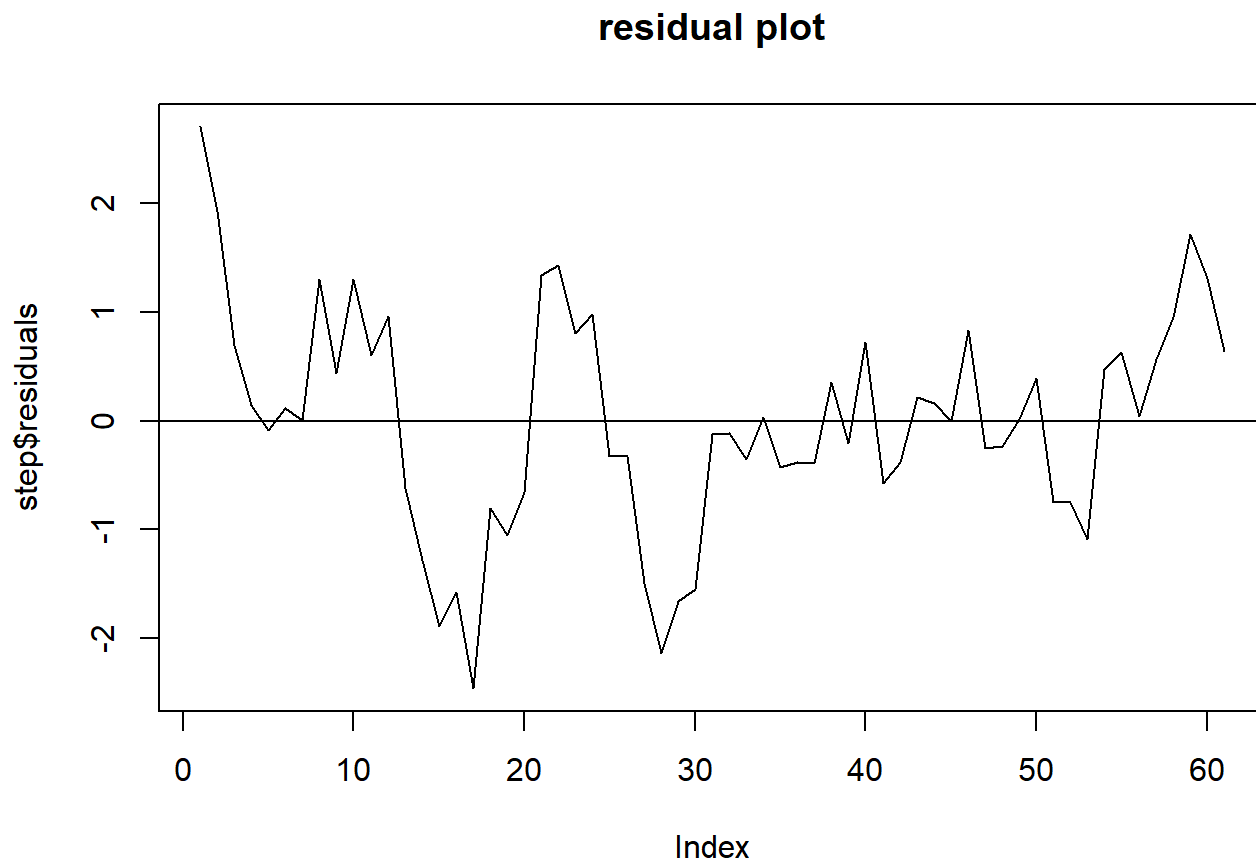
```
null <- lm(ts ~ 1)  
full <- lm(ts ~ full_df$time * full_df$Volume + cos(2*pi*time) + sin(2 * pi* time),full_df)  
step <- stepAIC(null, scope = list(upper = full), direction="both")
```

```
## Start: AIC=107.24
## ts ~ 1
##
##           Df Sum of Sq   RSS   AIC
## + full_df$time      1  273.192  69.25  11.735
## + sin(2 * pi * time) 1   53.180 289.26  98.943
## + full_df$Volume     1   19.142 323.30 105.729
## <none>                 342.44 107.238
##
## Step: AIC=11.73
## ts ~ full_df$time
##
##           Df Sum of Sq   RSS   AIC
## + full_df$Volume     1    6.757  62.49   7.471
## <none>                 69.25  11.735
## + sin(2 * pi * time) 1    0.756  68.49  13.065
## - full_df$time       1  273.192 342.44 107.238
##
## Step: AIC=7.47
## ts ~ full_df$time + full_df$Volume
##
##           Df Sum of Sq   RSS   AIC
## <none>                 62.49   7.471
## + sin(2 * pi * time)   1    0.928  61.56   8.559
## + full_df$time:full_df$Volume 1    0.585  61.90   8.898
## - full_df$Volume       1    6.757  69.25  11.735
## - full_df$time         1  260.807 323.30 105.729
```

```
summary(step)
```

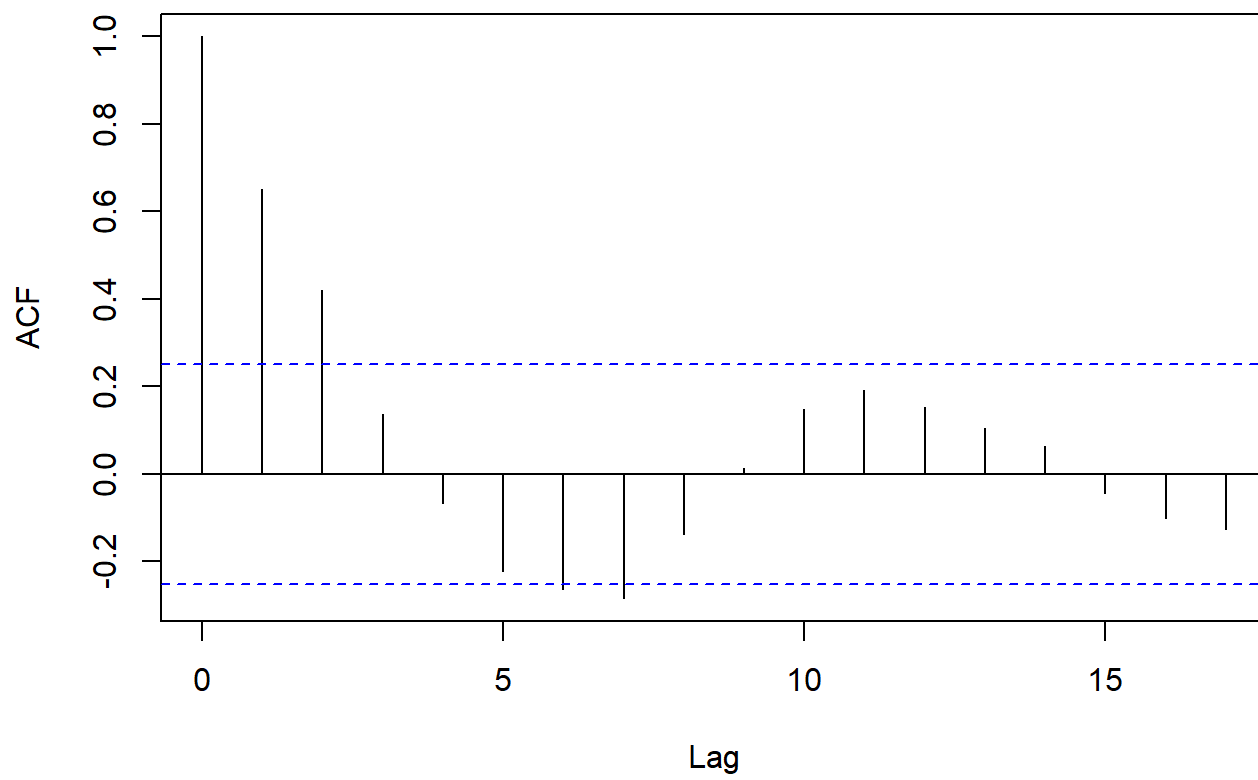
```
##
## Call:
## lm(formula = ts ~ full_df$time + full_df$Volume)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.45999 -0.56914  0.00596  0.64370  2.70821
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.607e+01  5.341e-01 104.981  <2e-16 ***
## full_df$time   1.181e-01  7.593e-03  15.559  <2e-16 ***
## full_df$Volume -3.510e-06  1.402e-06  -2.504   0.0151 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.038 on 58 degrees of freedom
## Multiple R-squared:  0.8175, Adjusted R-squared:  0.8112
## F-statistic: 129.9 on 2 and 58 DF,  p-value: < 2.2e-16
```

```
plot(x=step$residuals, type='l', main="residual plot")  
abline(h=0)
```



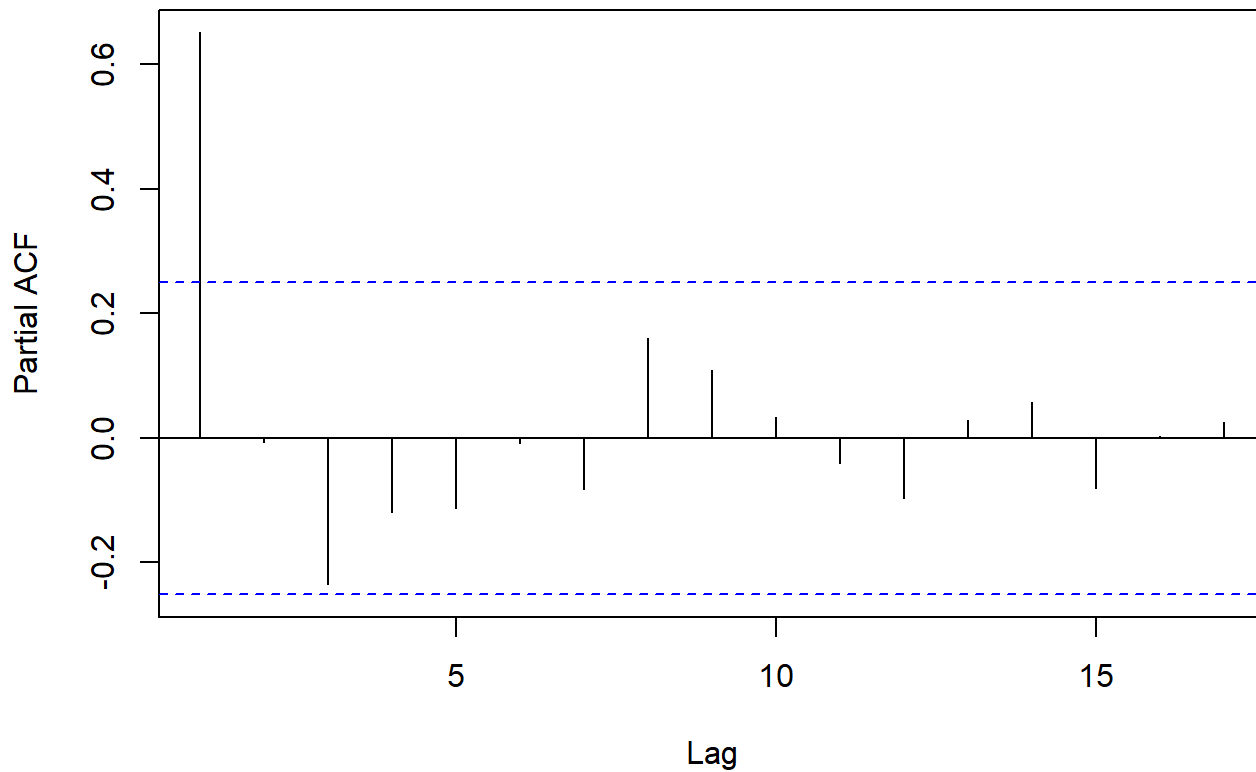
```
acf(step$residuals, main="ACF plot")
```

ACF plot



```
pacf(step$residuals, main="PACF plot")
```

PACF plot



This looks like AR(1): $x_t = \phi \cdot x_{t-1}$

find phi

```
acvf <- acf(step$residuals, type="covariance", plot=FALSE)
```

```
pacf_res <- pacf(step$residuals, plot=FALSE)
```

```
model <- ar.yw(step$fitted.values)
model$x.mean # mean estimate
```

```
## [1] 58.64197
```

```
model$ar # phi1 and phi2 estimates
```

```
## [1] 0.9414489
```

```
sqrt(diag(model$asy.var.coef)) # their standard errors
```

```
## [1] 0.04389394
```

```
model$var.pred # error variance estimate
```

```
## [1] 0.5393713
```

```
# forecast
```

```
model_pred <- predict(model, n.ahead = 30)
```

```
U <- model_pred$pred + model_pred$se
```

```
L <- model_pred$pred - model_pred$se
```

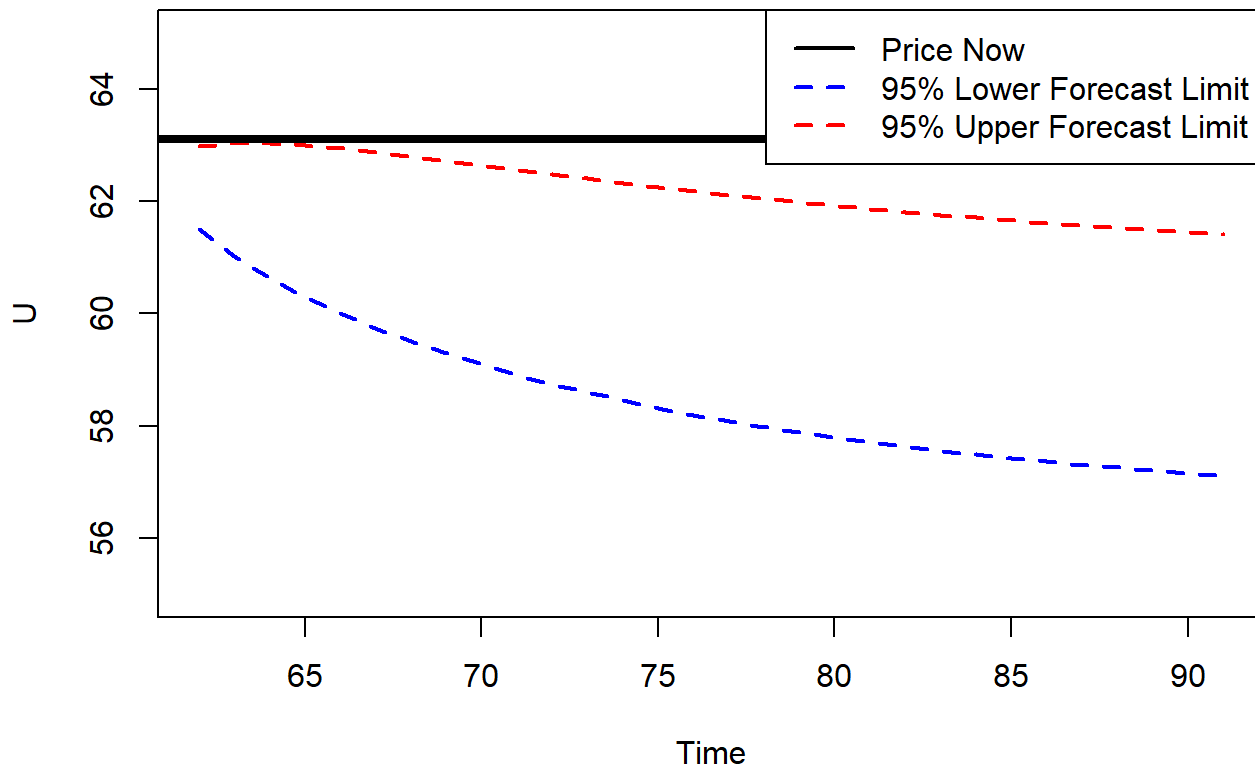
```
plot(U, col="red", lty="dashed", ylim=c(55, 65), lwd=2, main="Forecast")
```

```
lines(L, col="blue", lty="dashed", lwd=2)
```

```
abline(h = tail(full_df$combined,1), col="black", lwd=4)
```

```
legend("topright", legend = c("Price Now", "95% Lower Forecast Limit", "95% Upper Forecast Limit"), col = c("black", "blue", "red"), lty = c(1,2,2), lwd = 2)
```

Forecast



Testing ARMA (1,0)

```

for(i in 1:4){
  full_df <- data.frame(DAL, CLF)
  full_df$combined <- c((DAL$Close + CLF$Close) / 2)
  full_df$relative_strength <- ((DAL$Close + CLF$Close)/2 - (DAL$Low + CLF$Low)/2) / ((DAL$High +
CLF$High) / 2 - (DAL$Low + CLF$Low)/2)
  full_df <- full_df[(length(full_df$Close) - 120):length(full_df$Close) - 60,]
  test_df <- full_df[(length(full_df$Close) - 60):length(full_df$Close),]
  ts <- ts(full_df$combined)
  time <- time(ts)
  full_df$ts <- ts
  full_df$time <- time

  model_pred <- predict(model, n.ahead = 30)
  U <- model_pred$pred + model_pred$se
  L <- model_pred$pred - model_pred$se
  plot(U, col="red", lty="dashed", ylim=c(55, 65), lwd=2, main="120 days ago vs 60 days ago")
  lines(L, col="blue", lty="dashed", lwd=2)
  abline(h = 58.095, col="black", lwd=4)
  legend("topright", legend = c("Day 21 Price", "95% Lower Forecast Limit", "95% Upper Forecast L
imit"), col = c("black", "blue", "red"), lty = c(1,2,2), lwd = 2)
}

```

120 days ago vs 60 days ago

