STAT 520 – Homework 5 – Fall 2023

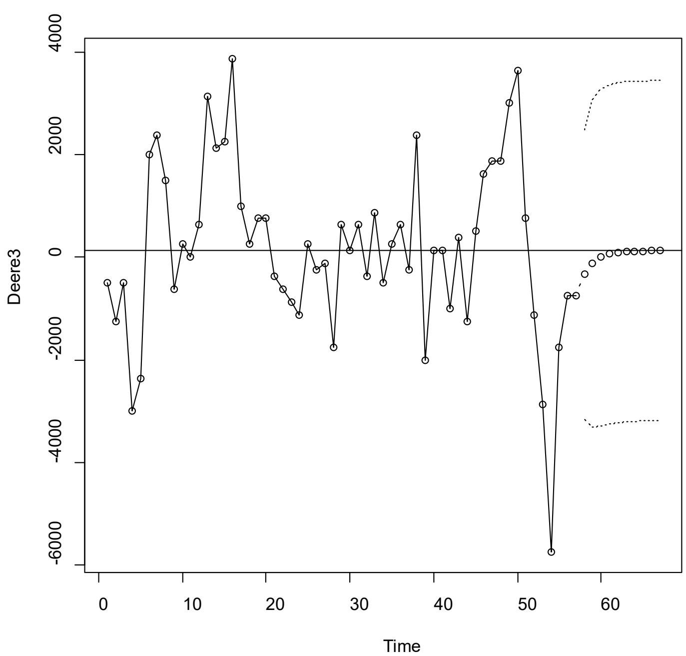
Note: For Homework 5A, do Problems 1 and 2 below.

Homework 5B consists of Problems 3 and 4, but note that Problem 3 is mandatory for graduate students and extra credit for undergraduates.

1) A data set of 57 consecutive measurements from a machine tool are in the `deere3` object in the `TSA` package. Type `library(TSA); data(deere3); print(deere3)` in R to see the data set.
(a) Fit an AR(1) model and use it to forecast the next ten values of the series, and list the forecasted values. Also plot the series along with the forecasted values and 95% prediction limits for the next ten values of the series.

$(Y_t - 124.38) = 0.526(Y_{t-1} - 124.38) + e_t$

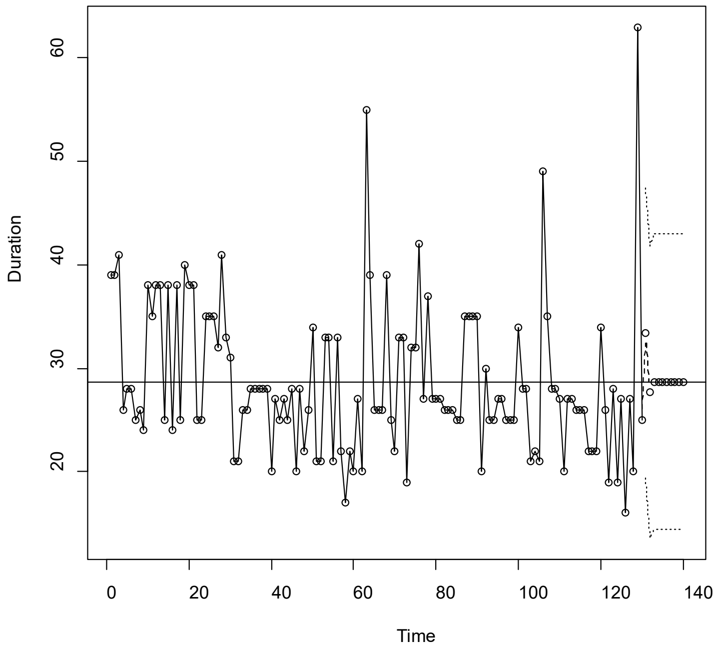
Forecasts for next 10 values: -335.145915 -117.120755 -2.538371 57.680013 89.327581 105.959853 114.700888 119.294709 121.708976 122.977786



2) A data set of durations until payment for 130 consecutive orders from a Winegrad distributor are in the `days` object in the `TSA` package. Type `library(TSA); data(days); print(days)` in R to see the data set.
(a) Fit an MA(2) model and use it to forecast the next ten values of the series, and list the forecasted values. Also plot the series along with the forecasted values and 95% prediction limits for the next ten values of the series.

$Y_t = 28.69 + e_t + 0.111e_{t-1} + 0.156e_{t-2}$

Forecast for next 10 values: 33.43453 27.67666 28.69310 28.69310 28.69310 28.69310 28.69310 28.69310 28.69310 28.69310



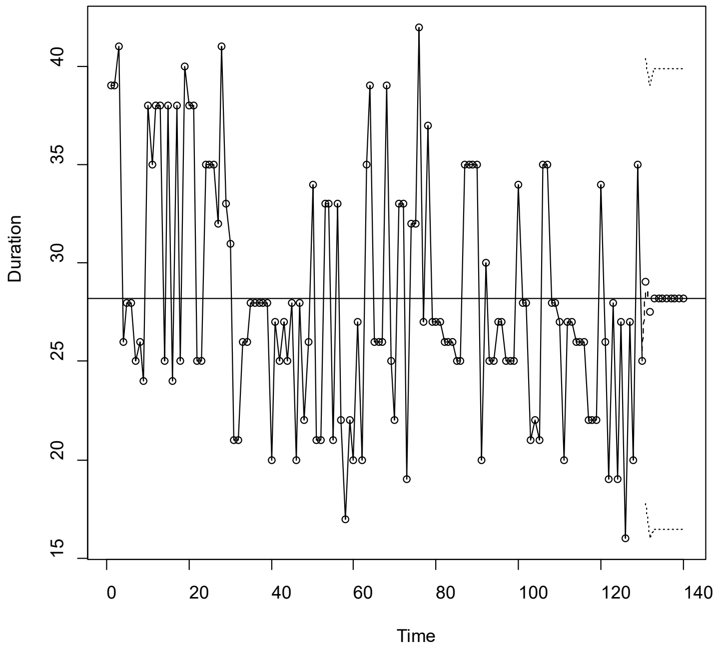
(b) There are three clear outliers (at times 63, 106, and 129) which can be seen from the time series plot. Replace these outliers with a "typical duration" value of 35. To do this, you can use the code

```
days.adj=days; days.adj[c(63,106,129)]=35; print(days.adj)
```

Fit an MA(2) model to this adjusted series, and use it to forecast the next ten values of the series, and list the forecasted values. Also plot the series along with the forecasted values and 95% prediction limits for the next ten values of the series.

$Y_t = 28.20 + e_t + 0.189e_{t-1} + 0.196e_{t-2}$

Forecast for next 10 values: 29.07436 27.52056 28.19564 28.19564 28.19564 28.19564 28.19564 28.19564 28.19564 28.19564

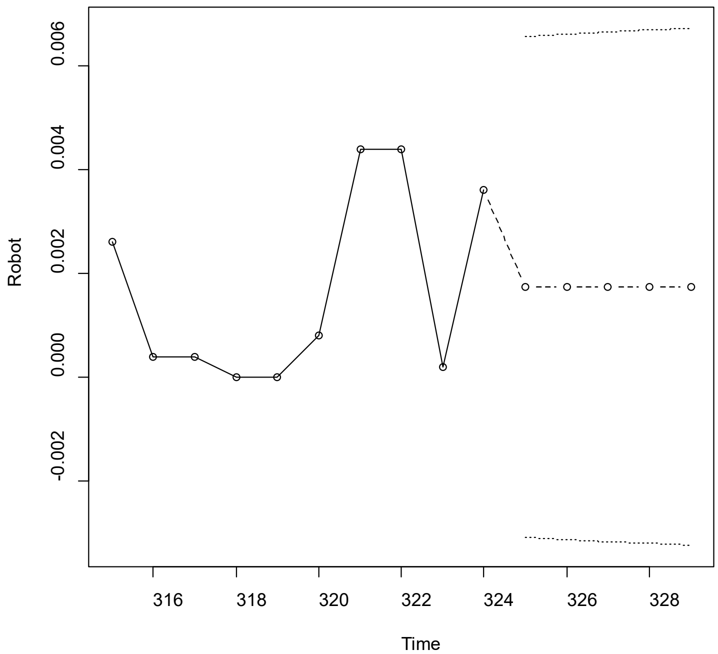


(c) Comment on any differences between the forecasts based on the original data and the forecasts based on the adjusted data.

Both forecasts converge toward the sample mean as the lead time increases, but this sample mean is slightly lower for the adjusted data set. The forecast for one time unit ahead is substantially higher using the original data.

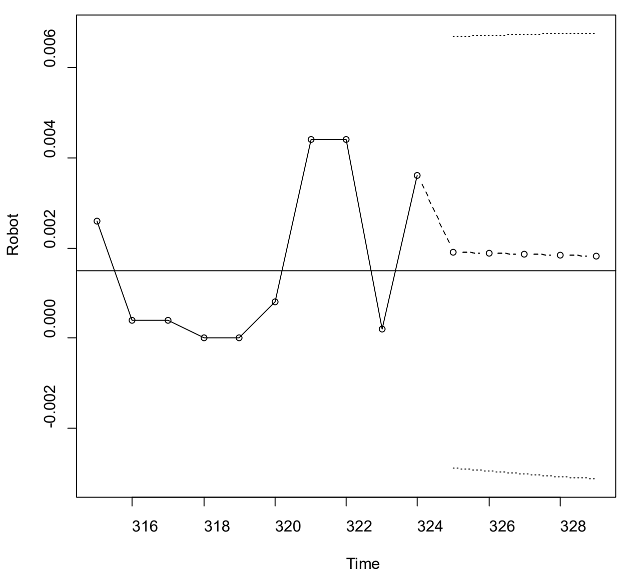
3) A data set of 324 measurements of an industrial robot's positions are in the `robot` object in the `TSA` package. Type `library(TSA); data(robot); print(robot)` in R to see the data set.
(a) Fit an IMA(1,1) model and use it to forecast the next five values of the series, and list the forecasted values. Also plot *the last ten observed values of the series* along with the forecasted values and 95% prediction limits for the next five values of the series. [Hint: type `help(plot.Arima)` and look at the `n1` argument of the `plot` function.]

Forecasts for next 5 values: 0.001742672 0.001742672 0.001742672 0.001742672 0.001742672



(b) Fit an ARMA(1,1) model and use it to forecast the next five values of the series, and list the forecasted values. Also plot *the last ten observed values of the series* along with the forecasted values and 95% prediction limits for the next five values of the series.

Forecasts for next 5 values: 0.001901348 0.001879444 0.001858695 0.001839041 0.001820424



(c) Compare the results from parts (a) and (b).

The forecasted values are a bit higher using the ARMA(1,1) model in (c) compared to the IMA(1,1) model in (b).

4) A data set of monthly electricity generation values are in the `electricity` object in the `TSA` package. Type `library(TSA); data(electricity); print(electricity)` in R to see the data set.

(a) Fit a seasonal means model which also contains a linear time trend on the (natural) *logarithms* of the data. [Some Chapter 10 example R code on the course web page may help with this.] Use it to forecast the next three months of the series (January 2006, February 2006, and March 2006), and list the forecasted values. [Hint: This is just a deterministic model; no ARIMA-type forecasting is needed.]

```
> #Predicted January 2006 logged value:
>
> 2.526e-02*2006-3.783e+01
[1] 12.842
>
> #or equivalently:
> future.time<-2006
> as.numeric( coef(model.s) %*% c(future.time,1,0,0,0,0,0,0,0,0,0,0,0) )
[1] 12.839
>
> #Predicted Feb. 2006 logged value:
>
> 2.526e-02*2006.083-3.795e+01
[1] 12.724
>
> #or equivalently:
> future.time<-2006.083
> as.numeric( coef(model.s) %*% c(future.time,0,1,0,0,0,0,0,0,0,0,0,0) )
[1] 12.716
>
> #Predicted March 2006 logged value:
>
> 2.526e-02*2006.167-3.792e+01
[1] 12.756
>
> #or equivalently:
> future.time<-2006.167
> as.numeric( coef(model.s) %*% c(future.time,0,0,1,0,0,0,0,0,0,0,0,0) )
[1] 12.752
```

(b) Give the forecasted values for the next three months in terms of the original data, i.e., not in logged values.

```
> #Predicted January 2006 value:
>
> exp(2.526e-02*2006-3.783e+01)
[1] 377588
>
> #or equivalently:
> future.time<-2006
> exp(as.numeric( coef(model.s) %*% c(future.time,1,0,0,0,0,0,0,0,0,0,0,0) ))
[1] 376499
>
> #Predicted Feb. 2006 value:
>
> exp(2.526e-02*2006.083-3.795e+01)
[1] 335594
>
> #or equivalently:
> future.time<-2006.083
> exp(as.numeric( coef(model.s) %*% c(future.time,0,1,0,0,0,0,0,0,0,0,0,0) ))
[1] 333081
>
> #Predicted March 2006 value:
>
> exp(2.526e-02*2006.167-3.792e+01)
[1] 346549
>
> #or equivalently:
> future.time<-2006.167
> exp(as.numeric( coef(model.s) %*% c(future.time,0,0,1,0,0,0,0,0,0,0,0,0) ))
[1] 345273
```