# Exam 4 (Summer 2024)

Math 531T: Time Series Analysis and Forecasting

Due Saturday, June 29, 2024

## Question 1

The dataset UKg2024 provides the log natural gas consumption in the UK from June 1980 to June 1984. The consumption is measured in millions of tons of coal equivalent.

a. Upload the UKgas in RStudio. Be sure to create the appropriate time structure. Use July 1980 to June 1984 as for your modelling (training) data and leave the last 12 observations aside (July 1984 - June 1985) for prediction/validation later in your analysis. Perform a SHORT exploratory analysis on the time series for July 1980 - June 1984. Comment on any general and seasonal trends you see in the data. Use the decompose function to obtain residuals and assess stationarity.

b. From part a, we can see that there is a strong monthly seasonality. Let's try fitting a linear factor model on the centered data (see below). Print out the model summary and perform a residual analysis to check model assumptions. Do assumptions appear to be met? Comment on any arma orders (p,q) that you think we may need to consider for modeling any autocorrelations left in the residuals (you may run the auto.arima() on the residuals to confirm your suspicions).

```
#Note that our first month is actually July
X <- as.factor(c(rep(seq(1:12),4)))

cen.mdat <- c(scale(modelling.data, center = TRUE, scale = FALSE))
mod.lm <- lm(cen.mdat ~ X - 1)
```

c. It seems that an MA(2) would capture the autocorrelation structure. Apply and ARMA(0,2) to the residuals of the lm model and confirm that indeed the residuals are now independent of one another.

d. Now, let's investigate how to fit this model in the dlm package. First, let's look at the seasonality. Print out the dlmModSeas(12) and provide the F, G, and W structures for each. What might the state vector for this model be representing? Let's explore by filtering the seasonal dlm model using the estimated $\hat{\sigma}^2$ from the lm model:

```
seas.dlm.model = dlmModSeas(12,dV=(0.4241^2),dW=rep(0,11))

seas.dlm.filter = dlmFilter(cen.mdat, seas.dlm.model)
n = length(cen.mdat) + 1 # Due to theta_0
seas.dlm.filter$m[n,] #
```

d. (cont) You should see these values match of with the monthly effect estimates from the lm model, but in reverse order and missing the first month (which is July in our data). Let's fix this (see code below). Then write a couple sentences explaining how the G matrix in the dlmModSeas operates on the state vector to extract the monthly effects.

```
data.frame(mean = c(-sum(seas.dlm.filter$m[n,]), rev(seas.dlm.filter$m[n,])), lm_mean = mod.lm$coefficients)
```

e. Alright, now having understanding of the interpretation behind dlmModSeas(12) let's model the Seasonality and ARMA components together in the dlm package by running the following (provide the values of your parameter estimates - what is odd about the ma coefficients? Explain.):

```
model1.build <- function(parm) {

    #Seasonal Term
```

```
     #Season Factor model
     season <- dlmModSeas(frequency = 12,
               dV=(0.0001), dW=rep(0,11))
    # ARMA Term
    arma <- dlmModARMA(ma = parm[1:2],
                       sigma2 = exp(parm[3]))
    return(season + arma)
}

model1.mle <- dlmMLE(cen.mdat, parm=c(0,0,0), build=model1.build)

dlm.mod1 <- model1.build(model1.mle$par)
model1.mle$par
```

f. Let's see how well we do in forecasting. I provide a code below, but feel free to modify (especially if you have a preferred syntax), but be sure to make analogous plots: Plot1: provide the full centered dataset, smoothed model fit, forecast estimates. Plot2: provide vallidation data, and forecasts with interval estimates. Does the model captures the observed vallidation data?

```
model1.filtered <- dlmFilter(cen.mdat, dlm.mod1)

model1.smoothed <- dlmSmooth(cen.mdat, dlm.mod1)

f1 <- c(model1.smoothed$s[-1,]%*%t(FF(dlm.mod1)))

model1.forecast <- dlmForecast(model1.filtered, nAhead=12, sampleNew = 1000)

a1 <- drop(model1.forecast$a%*%t(FF(dlm.mod1)))

pred1 <- matrix(unlist(model1.forecast$newObs), ncol = 12, byrow = TRUE)

low1 <- apply(pred1, 2, quantile, probs =0.025)
high1 <- apply(pred1, 2, quantile, probs =0.975)

cen.dat <- c(dat.ts -mean(dat.ts))

x <- index(dat.ts) #zoo library
df.comp <- rbind(
  data.frame(x=x, y=cen.dat, series="data"),
 data.frame(x=x[49:60], y=a1, series="DLM forecast"),
 data.frame(x=x[1:48], y=f1, series="DLM Smoothed Fit")

  )

g.comp <- ggplot(subset(df.comp, x>1980), aes(x=x, y=y, colour=series)) + geom_line(aes(linetype =series))

g.comp
```

```
df1.comp <- rbind(
  data.frame(x=x[49:60], y=cen.dat[49:60], series="data"),
 data.frame(x=x[49:60], y=a1, series="DLM forecast"),
 data.frame(x=x[49:60], y=low1, series="Lower"),
 data.frame(x=x[49:60], y=high1, series="Upper")

  )
g1.comp <- ggplot(subset(df1.comp, x>1984), aes(x=x, y=y, colour=series)) + geom_line(aes(linetype =series))

g1.comp
```

g. Fit a SARIMA$(0, 0, 2)(0, 1, 0)[12]$ to the centered modelling data, and compare model ma estimates and performance to the dlm Seas + ARMA model. Discuss your results and provide any relevant plots.

## Question 2

Let's explore the same dlm Seas + ARMA model in the case we had some observations missing. You can work directly with the centered data. Omit 8 observations of your choice, for example, below I omit some values in 1983 and 1984

```
cen.dat.inc <- cen.dat
# set some missing values
cen.dat.inc[c(33:36,48:51)] <- NA

temp <- cbind(x, cen.dat.inc)
```

Re-build your dlm using the incomplete data and obtain the Filtered and Smoothed imputation of the missing values. Report the sum of square errors of the imputed values with actual values. Comment on the model's imputation performance.