

# Lab 3 - The EDA of TSA

Benjamin M. Taylor, Kevin Hayes

January 20, 2023

## 1 Smoothing COVID-19 Counts

Data concerning the progression of the COVID-19 pandemic across the globe can be obtained from the website [https://www.worldometers.info/coronavirus/]. A time-series plot of the count of New Cases per Day of the virus in Ireland can be found at [https://www.worldometers.info/coronavirus/country/ireland/#graph-cases-daily/]. Beneath the plot the user is provided with the option of adding a 3-day moving average or a 7-day moving average or both. These have the effect of smoothing over the jagged daily counts and allow us to eyeball the general trend.

```
[ ]: options(repr.plot.width=15, repr.plot.height=15) # makes plots bigger in the ↵  
      ↪ webpage
```

## 2 LA Pollution-Mortality Study

Shumway and Stoffer (2011) discussed data of the average weekly cardiovascular mortality in Los Angeles County; 508 six-day smoothed averages obtained by filtering daily values over the 10 year period 1970-1979. The data can be accessed in the R package `astsa`. Execute the following R commands.

```
[ ]: require(astsa)
```

Have a look at the help file for the dataset `lap`:

`?lap`

```
[ ]: par(mfrow=c(3,1))  
     plot(cmort, main="Cardiovascular Mortality", xlab="", ylab="")  
     plot(tempr, main="Temperature", xlab="", ylab="")  
     plot(part, main="Particulates", xlab="", ylab="")  
     # kill plot before moving on
```

## 3 Regression Analysis

Let's consider deploying some very basic regression models on our time-series data. Execute the following R commands.

```
[ ]: pairs(cbind(Mortality=cmort, Temperature=tempr, Particulates=part))
```

```
[ ]: # Regression
temp = tempr-mean(tempr) # center temperature
temp2 = temp^2 # square it
trend = time(cmort) # time

fit = lm(cmort~ trend + temp + temp2 + part, na.action=NULL)

[ ]: summary(fit) # regression results

[ ]: summary(aov(fit)) # ANOVA table (compare to next line)

[ ]: summary(aov(lm(cmort~cbind(trend, temp, temp2, part)))) # Table 2.1

[ ]: num = length(cmort) # sample size
AIC(fit)/num - log(2*pi) # AIC

[ ]: BIC(fit)/num - log(2*pi) # BIC

[ ]: # AIC(fit, k=log(num))/num - log(2*pi) # BIC (alt method)
(AICc = log(sum(resid(fit)^2)/num) + (num+5)/(num-5-2)) # AICc
```

## 4 Moving Average Smoothers

Execute the following R commands. Pay attention to the architecture of the linear filters (moving average smoother) being used.

```
[ ]: ma3 = filter(cmort, sides=2, rep(1,2)/2)
ma2.2 = filter(cmort, sides=2, c(1,2,1)/4)

[ ]: plot(cmort, type="p", ylab="mortality", main="Cardiovascular Mortality")
lines(ma3, col="red")
lines(ma2.2, col="blue")
```

## 5 COSINOR Regression

How about a classical regression model based on cosine and sine predictors.

```
[ ]: #Example 2.11
wk = time(cmort) - mean(time(cmort)) # wk is essentially t/52 centered at z
wk2 = wk^2
wk3 = wk^3
cs = cos(2*pi*wk)
sn = sin(2*pi*wk)
reg1 = lm(cmort~wk + wk2 + wk3, na.action=NULL)
reg2 = lm(cmort~wk + wk2 + wk3 + cs + sn, na.action=NULL)
```

```
[ ]: plot(cmort, type="p", ylab="mortality")
      lines(fitted(reg1))
      lines(fitted(reg2))
```

## 6 Smoothing Methods

Other smoothing methods we might consider include: kernel smoothing; super smoother, and lowess smoothers; smoothing splines; the kitchen sink.

```
[ ]: #Example 2.12
      plot(cmort, type="p", ylab="mortality")
      lines(ksmooth(time(cmort), cmort, "normal", bandwidth=5/52))
      lines(ksmooth(time(cmort), cmort, "normal", bandwidth=2))
```

```
[ ]: #Example 2.13
      par(mfrow=c(2,1))
      plot(cmort, type="p", ylab="mortality", main="nearest neighbour")
      lines(supsmu(time(cmort), cmort, span=.5))
      lines(supsmu(time(cmort), cmort, span=.01))

      plot(cmort, type="p", ylab="mortality", main="lowess")
      lines(lowess(cmort, f=.02))
      lines(lowess(cmort, f=2/3))
```

```
[ ]: #Example 2.14
      plot(cmort, type="p", ylab="mortality")
      lines(smooth.spline(time(cmort), cmort))
      lines(smooth.spline(time(cmort), cmort, spar=1))
```

```
[ ]: #Example 2.15
      par(mfrow=c(2,1), mar=c(3,2,1,0)+.5, mgp=c(1.6,.6,0))

      plot(tempr, cmort, main="lowess", xlab="Temperature", ylab="Mortality")
      lines(lowess(tempr, cmort))

      plot(tempr, cmort, main="smoothing splines", xlab="Temperature", ylab="Mortality")
      lines(smooth.spline(tempr, cmort))
```

## 7 Multiple Linear Regression: Automated Model Search and Diagnostic Plots

The next dataset has one response variable and nine explanatory variables (get data here [Download here](#)). These are not time series data, rather a classical dataset that we will use to demonstrate various information criteria used with automatic model fitting. First of all, we will enter the data into R. The following few lines organizes the data in a 8406 by 10 matrix, and the convert this to a data frame.

```
[ ]: dat = matrix(scan(file="data/nineXvars.txt"),ncol=10,byrow=TRUE)
      dat = as.data.frame(dat)
      names(dat) = c("y", paste("X",1:9, sep=""))
```

```
[ ]: dim(dat)
```

```
[ ]: names(dat)
```

```
[ ]: pairs(dat)
```

We start by fitting a very large model, that includes interaction terms and quadratic terms between and for all nine explanatory variables.

```
[ ]: fmla = as.formula( paste("y ~ .^2 +", paste("I(X",1:9,"^2)", sep="", collapse="␣
      ↪+ ") ))
      fmla
```

```
[ ]: mod = lm( fmla , data = dat)

      summary(mod) ## overview of linear model
```

Next, we use R to search the model space. Starting from this very large model we will use the R function step to implement a backwards search strategy using the AIC as the variable rejection criterion.

```
[ ]: mod.aic = step(mod, direction = "backward", k = 2)
```

```
[ ]: summary(mod.aic)
```

The AIC is conservative in the sense that it retains more variable than maybe needed. An alternative criterion is the BIC “Bayesian Information Criterion.”

```
[ ]: mod.bic = step(mod, direction = "backward", k = log(nrow(dat)))
```

```
[ ]: summary(mod.bic)
```

Finally, it is always good practice to plot the residuals. Try the following R command. . .

```
[ ]: plot(fitted(mod.bic), residuals(mod.bic), pch = ".")
```

What is Homer telling us??