Workshop 1: Welcome!

STAT 464/864 Discrete Time Series Analysis

Skye P. Griffith, Queen's University

Markdown

"Markdown" is a simple syntax that allows you to stylize your document and its text entirely through the use of symbols.

> Block quote
>
> …block quote continued

Italic font

Bold font

Use 1 to 4 hashtags to make a **header.** More hashtags = *smaller* header! You'll need to use headers to organize your assignments and reports for this class.

Example: 1 Say problem 1 has parts a), b) and c)

Problem 1

Part a)

Put your answer to part **b)** here.

Part b)

Put your answer to part **b)** here.

Part c)

Put your answer to part **b)** here.

Chunks

To include R code in your document, you'll need to distinguish it from text. To do this, create a "chunks" using the back-tick ‘ character (same key as the ∼ key on your keyboard).

Example 2

Let's generate 10 observations from a $\mathcal{N}(0, 1)$ distribution, then print and plot their values.

> *The values will be printed, and the plot will appear as a figure in our final document.*

```r
# Comments are hashtags, just like regular R :)
x <- rnorm(10)

x        # This will print the values of x to your document
```

```
 [1] -0.1337931  0.6547342 -0.6246319  0.3050322 -1.7748245 -1.7132225
 [7]  0.5646987 -0.2940028  1.5519755  0.1024572
```

```r
plot(x)  # This will produce the plot in figure 1.
```

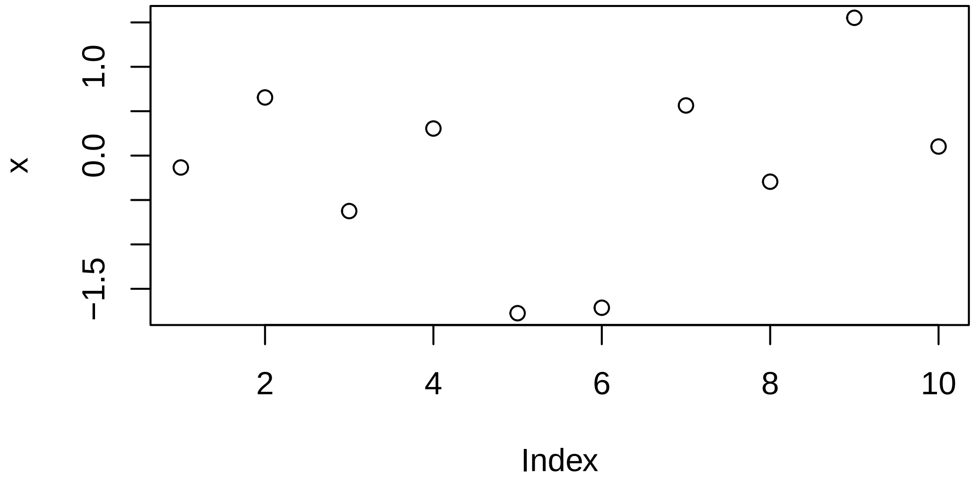


Figure 1: A plot generated by Quarto, as per the code in example 2

That's the whole reason we're doing this. The content of your homework submissions will be the output of your actual code. It will *update* every time you render the document (this is called **dynamic output**). That means, if you're generating **random data,** your plots will look different every time. It also means you can't have any typos in your code, or else your document won't render!

You can also create LaTeX chunks to type math equations.

$$E[X] = \frac{1}{N}\sum_{i=1}^{N} x_i$$

Working directory

So… you type in a .qmd file, you produce a .pdf file, and a bunch of weird files are generated and erased during the rendering process. We're about to start using images and datasets, too. You're going to want to store all this stuff in the same folder: this will be your *working directory.*

I like to make a new folder (directory) for each project I work on, but if you want, you can get away with making a single folder for this course. Once you have your desired folder, the easiest thing to do is to create an R project (file -> new R project) inside that folder.

You can set your working directory by clicking the gear in the RStudio "files" pane, or, you can set it manually like this:

```r
setwd("C:/Users/skyep/Desktop/464/TimeSeries_FA24/Workshop_Files/Workshop_1")
```

Inserting images

If you're not familiar with LaTeX, you probably don't want to type out all the actual *math* in your homework solutions. Save your handwritten solutions as a .png file **in your working directory.**

Example 3:

Upload your handwritten solution for part a) of some homework problem

Part a)

The exponential smoother is:

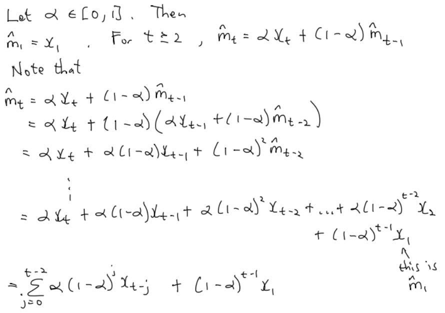
Let $\alpha \in [0,1]$. Then

$\hat{m}_1 = x_1$. For $t \geq 2$, $\hat{m}_t = \alpha x_t + (1-\alpha)\hat{m}_{t-1}$

Note that

$$
\begin{aligned}
\hat{m}_t &= \alpha x_t + (1-\alpha)\hat{m}_{t-1} \\
&= \alpha x_t + (1-\alpha)\left(\alpha x_{t-1} + (1-\alpha)\hat{m}_{t-2}\right) \\
&= \alpha x_t + \alpha(1-\alpha)x_{t-1} + (1-\alpha)^2\hat{m}_{t-2} \\
&\;\vdots \\
&= \alpha x_t + \alpha(1-\alpha)x_{t-1} + \alpha(1-\alpha)^2 x_{t-2} + \dots + \alpha(1-\alpha)^{t-2}x_2 \\
&\qquad + (1-\alpha)^{t-1}\underset{\substack{\wedge \\ \text{this is} \\ \hat{m}_1}}{x_1} \\
&= \sum_{j=0}^{t-2}\alpha(1-\alpha)^j x_{t-j} + (1-\alpha)^{t-1}x_1
\end{aligned}
$$

You can also add captions to figures, if you want.



Let $\alpha \in [0,1]$. Then

$\hat{m}_1 = x_1$. For $t \geq 2$, $\hat{m}_t = \alpha x_t + (1-\alpha)\hat{m}_{t-1}$

Note that

$$
\begin{aligned}
\hat{m}_t &= \alpha x_t + (1-\alpha)\hat{m}_{t-1} \\
&= \alpha x_t + (1-\alpha)\left(\alpha x_{t-1} + (1-\alpha)\hat{m}_{t-2}\right) \\
&= \alpha x_t + \alpha(1-\alpha)x_{t-1} + (1-\alpha)^2\hat{m}_{t-2} \\
&\;\vdots \\
&= \alpha x_t + \alpha(1-\alpha)x_{t-1} + \alpha(1-\alpha)^2 x_{t-2} + \dots + \alpha(1-\alpha)^{t-2}x_2 \\
&\qquad + (1-\alpha)^{t-1}\underset{\substack{\wedge \\ \text{this is} \\ \hat{m}_1}}{x_1} \\
&= \sum_{j=0}^{t-2}\alpha(1-\alpha)^j x_{t-j} + (1-\alpha)^{t-1}x_1
\end{aligned}
$$

Figure 2: caption goes here

Importing Data

Randomly generated data is fun and all that, but we'll mostly want to use *real data* in our documents. Save your desired dataset **in your working directory,** and then load it like so:

```
my.wine <- read.csv("wine1.csv",       # data file (we'll usually use .csv files)
                    header = FALSE)    # first row = observations

colnames(my.wine) <- c("Date","Sales") # Name the dataset's variables
```

And here we can plot the data:

```
par(mar = c(4,4,2,1)) # (I usually include this line of code before any plot)

plot(my.wine$Sales,
     main = "Austrailian Red Wine Sales",
     xlab = "Time (see next workshop for units)",
     ylab = "Sales in thousand liters",
     type = "l")
```

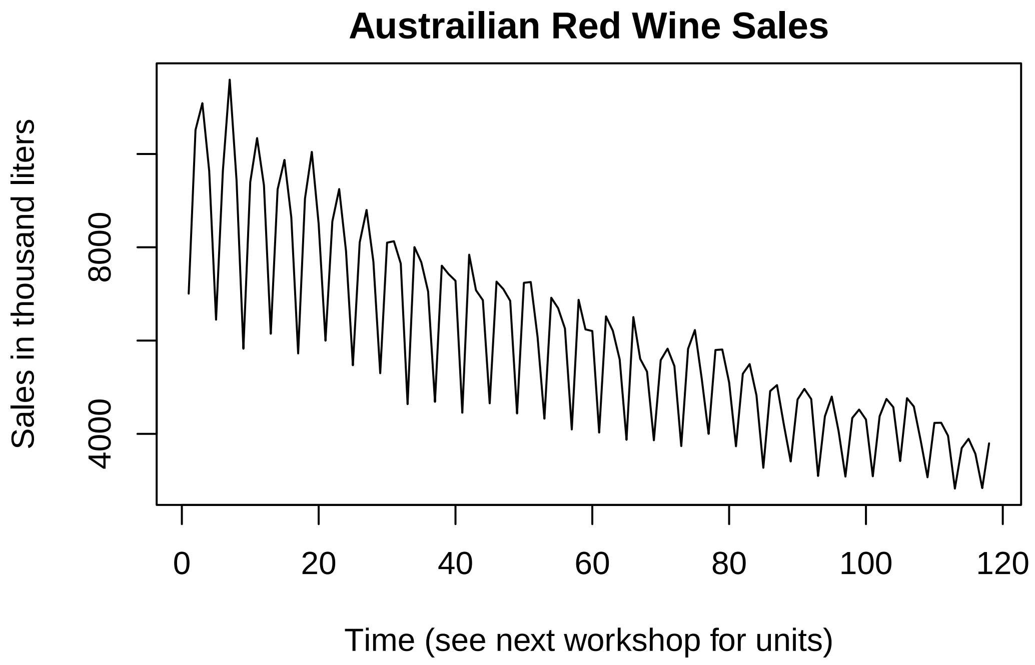


Figure 3: *Imported* wine data, pardon the pun

Tips, Tricks, and Troubleshooting

We'll go over some of these if we get a chance to. If not, I recommend you read this section on your own time.

Inline code

Recall that we generated the variable `x` randomly. We can display those values inline by creating a mini-chunk, which is also capable of dynamic output.

> *The 5th observation in the vector* `x` is -1.7748245

Knitting errors

- Documents always render based on an **empty environment.** Imagine you are sending your code to a friend and they have to run it from scratch.
- Try knitting *every time you finish a new chunk.* Debugging one chunk at a time is easiest, especially when you're learning.
- Name your chunks so you can find them easier
- *Read* the error text: what *line* does Quarto say the error is located? What *chunk?*

Formatting errors

To make sure your markdown syntax doesn't bug out, leave one line of **vertical space** between:

- Headers
- Chunks
- Paragraphs
- LaTeX Equation environments (double dollar signs)
- Lists (like this one) but not list *items*

Creating a Table

A	B	C	D	E	F
a1	b1	c1	d1	e1	f1
a2	b2	c2	d2	e2	f2
a3	b3	c3	d3	e3	f3
a4	b4	c4	d4	e4	f4

Notice the alignment of the columns is controlled by the placement of the colon (:)