WolfWare / Dashboard / My courses / CSC 591 (603) SPRG 2019
/ Topic-4: Time Series Decomposition, Analysis, Forecasting
/ (DUE: 04/26/2019): SUBMIT: QUIZ: BONUS: Time Series Decomposition and Smoothing

Started on	Sunday, February 17, 2019, 5:18 PM
State	Finished
Completed on	Monday, May 13, 2019, 7:33 PM
Time taken	85 days 1 hour
Grade	**3.00** out of 34.00 (**9**%)

Question 1

Partially correct

3.00 points out of 4.00

Sam performed the following decompositional analysis of the monthly number of persons in the civilian labour force (read help(stl) and help(monthplot). Select all the conclusions that could be drawn from these two plots.

```
require(fpp)
plot(labour, main = "Monthly Civilian Labour Force")
fit <- stl(labour, s.window=12)
plot(fit)
monthplot(fit$time.series[,"seasonal"],
          ylab="Seasonal")
```

Select one or more:

- [x] a. The data has steady upward trend with small dips around 1991/1992 years. ✔
- [x] b. The data has rather strong seasonal pattern which changes over time. ✔
- [ ] c. The seasonal component changes mostly in March, then, in decreasing order, in September, April and December.
- [x] d. The recession of 1991/1992 visible in the trend as two dips and, in the remainder, as correlated values with higher than average amplitude. ✔

Your answer is partially correct.

You have correctly selected 3.

The correct answers are: The data has steady upward trend with small dips around 1991/1992 years., The data has rather strong seasonal pattern which changes over time., The seasonal component changes mostly in March, then, in decreasing order, in September, April and December., The recession of 1991/1992 visible in the trend as two dips and, in the remainder, as correlated values with higher than average amplitude.

Question 2

Not answered

Points out of 20.00

Anna analyzed sales data on one-family houses in US, using the R code snippet below. Specifically, she decomposed the time series into systematic and non-systematic parts: trend, seasonal, and remainder (using decompose() and stl() methods). Since stl() method only supports additive seasonality, she calculated seasonally adjusted data with seasadj() and ploted it on top of the original data to explore the effect of adjusting for additive seasonality.

```
require(fpp)
plot(decompose(hsales))
plot(stl(hsales,s.window="periodic"))
plot(stl(hsales,s.window=15))

plot(hsales,col="gray")
fit <- stl(hsales,s.window=15)
hsales.sa <- seasadj(fit)
lines(hsales.sa, col="red")
```

For this problem, you are asked to decompose the time series for the sales of plastic products (data(plastics)) and answer the following questions with the supporting R code to be included with the submission:

1. Plot the time series (ts) and comment on its trend (upward/downward, linear vs non-linear).
2. Decompose the ts using the decompose() method with the multiplicative seasonal component and plot the decomposed components. Do the decomposed components support your original conclusion from #1?
3. Are there any time periods where the trend could be non-linear?
4. Create seasonally adjusted series and plot it.
5. Modify your data by injecting the outlier (e.g., add some large value, say 500) in the middle of the data. Repeat steps 2-4 and comment on what effect i=this change had on either seasonal or trend components of the ts?
6. Repeat step #5 but this time add the outlier at the end of the ts.
7. Explain what the following code snippet is doing:

```
require(fpp)
fit <- decompose(plastics, type="multiplicative")
seasAdj <- seasadj(fit)
driftFit = rwf(seasAdj, drift=TRUE, h=24)
plot(driftFit)
plot(driftFit, ylim = c(500, 2200))
lines(driftFit$mean*fit$figure, col = "green", lwd = 2)
lines(driftFit$upper[,2]*fit$figure, col="red")
lines(driftFit$lower[,2]*fit$figure, col="red")
lines(plastics, col = "green", lwd = 2)
```

require(fpp)

plot(plastics, ylab = "Sales of product A")

Seasonal fluctuations have smooth sinusoidal shape. The data has upward, close to linear trend.

#—————————

fit <- decompose(plastics, type="multiplicative")

plot(fit)

The results partially support conclusions from the previous part. The trend appears to be nonlinear in the beginning and in the end.

#———

seasAdj <- seasadj(fit)

plot(seasAdj, ylab = "Seasonally adjusted data")

#—————

plastics2 = plastics

plastics2[31] = plastics2[31] + 500

fit <- decompose(plastics2, type="multiplicative")

plot(fit)

seasAdj <- seasadj(fit)

plot(seasAdj, ylab = "Seasonally adjusted data")

The outlier appears partially in the seasonally adjusted data.

#——————

plastics3 = plastics

plastics3[59] = plastics3[59] + 500

fit <- decompose(plastics3, type="multiplicative")

plot(fit)

seasAdj <- seasadj(fit)

plot(seasAdj, ylab = "Seasonally adjusted data")

The outlier apperas in the seasonally adjusted data almost completely. When outlier is at the end of the data, it affects the seasonal pattern less than when it is in the middle. Therefore, the outlier at the end of the data appears in the seasonally adjusted data more pronounced than when it is in the middle.

#---------

Random walk model with drift forecasts of the seasonally adjusted data is built and reseasonalized forecasts of the seasonally adjusted data are plotted

Question 3

Not answered

Points out of 10.00

Ann performed a simple 4-day exponential smoothing forecast for daily sales of paperback books using the ses() method, as depicted in the code snippet below. By setting the parameter to initial="simple", she explored different values of parameter α and its effect on the within-sample SSE (sum of squared errors), and found that the value of α approximately equal to 0.2 may give the best accuracy. She then changed her settings as initial="optimal" and let the ses() method select the optimal value of α. She finally discovered that although the values α for different initial settings vary significantly, the final forecasts were almost identical.

You are asked to perform a similar analysis but for the hardcover book series.

```
require(fpp)
data(books)
plot(books, main = "Data set books")
alpha = seq(0.01, 0.99, 0.01)
SSE = NA
for(i in seq_along(alpha)) {
  fcast = ses(books[,"Paperback"], alpha = alpha[i], initial = "si
  SSE[i] = sum((books[,"Paperback"] - fcast$fitted)^2)
}
plot(alpha, SSE, type = "l")
fcastPaperSimple = ses(books[,"Paperback"],
                       initial = "simple",
                       h = 4)
fcastPaperSimple$model$par[1]
plot(fcastPaperSimple)

fcastPaperOpt = ses(books[,"Paperback"],
                    initial = "optimal",
                    h = 4)
fcastPaperOpt$model$par[1]
plot(fcastPaperOpt)
as.numeric((fcastPaperOpt$mean -
  fcastPaperSimple$mean)/fcastPaperSimple$mean) * 100
```

* Parameter alpha affects accuracy of within-sample one-step forecasts.

* The best accuracy of within-sample one-step forecasts is achieved for alpha approximately equal to 0.35.

* Although the values alpha for different initial settings are rather close, the final forecasts are more distant than for paperback books case

◀ (DUE: 02/20/2019): SUBMIT: PROJECT: Bitcoin Price with Bayesian Regression

Jump to...

)19): SUBMIT: QUIZ: BONUS: TS Evaluation, Baseline Methods, White Noise, Stationarity, Stabilization ▶