Started on	Monday, February 18, 2019, 2:43 AM
State	Finished
Completed on	Monday, May 13, 2019, 8:20 PM
Time taken	84 days 16 hours
Grade	**0.00** out of 50.00 (**0**%)

Question 1

Not answered

Points out of 6.00

Paul examine the Dow-Jones index and concluded that it is non-stationary non-white noise time series data using the R code snippet below. He decided to stabilize the mean by applying the differencing. He augmented this code with the proper unit root test to determine the lag of differencing he should be performing on the data. He re-run the code below on the transformed (i.e., differenced) data and got very different results this time.

```
require(fpp)
data(dj)
plot(dj)
acf(dj)
Box.test(dj, lag=10, fitdf=0, type="Lj")
```

Submit the R code and the conclusions Paul might have drawn from these results.

```
differences <- ndiffs(dj)

plot(diff(dj, differences))

errors <- diff(dj,differences)

plot(errors)

acf(errors)

Box.test(errors, lag=10, fitdf=0, type="Lj")
```

Conclusions:

1. Because dj is non-stationary, then ndiffs() rather than nsdiffs() should be applied

2. After differencing, the means, has stabilized and the resulting time series became stationary

3. The resulting time series (errors) now looks like a white noise: no autocorrelations outside the 95% limits and Ljung-Box Q statistic has a p-value 0.15 for h = 10.

4. The obtained differences of the Dow-Jones index are the day-today changes. These daily change in the Dow-Jones index is essentially a random amount uncorrelated with previous days.

5. Random walk model could be applied to this non-stationary Dow-Jones index data, as it has long periods of apparent trends up or down with sudden and unpredictable changes in direction.

Question 2

Not answered

Points out of 20.00

For the following data sets find an appropriate Box-Cox transformation and order of differencing in order to obtain stationary data (data (package = "fma") and data(package="fpp")). Example transformations may include log(), diff(), double differencing via diff(diff(x)), and BoxCox(). There are multiple ways to transform a non-stationary ts to a stationary one. The provided feedback (available upon submission) is an example solution; others might be as equally valid.

1. usnetelec
2. usgdp
3. mcopper
4. enplanements
5. visitors

```
plot(usnetelec)

plot(diff(usnetelec))

Acf(diff(usnetelec), lag.max = 100, main = "")

#------------

Acf(ibmclose, lag.max = 100, main = "")

#---------

plot(usgdp)

Acf(usgdp, lag.max = 100, main = "")

lambda = BoxCox.lambda(usgdp)

plot(diff(BoxCox(usgdp, lambda)))

Acf(diff(BoxCox(usgdp, lambda)), lag.max = 100, main = "")

#---------------

plot(mcopper)

Acf(mcopper, lag.max = 100, main = "")

lambda = BoxCox.lambda(mcopper)

plot(diff(BoxCox(mcopper, lambda)))

#---------

plot(enplanements)
```

```
plot(log(enplanements))

Acf(log(enplanements), lag.max = 100, main = "")

plot(diff(log(enplanements), lag = 12))

Acf(diff(log(enplanements), lag = 12), lag.max = 100, main = "")

plot(diff(diff(log(enplanements), lag = 12)))

Acf(diff(diff(log(enplanements), lag = 12)), lag.max = 100, main = "")

#------------

plot(visitors)

lambda = BoxCox.lambda(visitors)

plot(BoxCox(visitors, lambda))

Acf(BoxCox(visitors, lambda), lag.max = 100, main = "")

plot(diff(BoxCox(visitors, lambda), lag = 12))

Acf(diff(BoxCox(visitors, lambda), lag = 12), lag.max = 100, main = "")

plot(diff(diff(BoxCox(visitors, lambda), lag = 12)))

Acf(diff(diff(BoxCox(visitors, lambda), lag = 12)), lag.max = 100, main = "")
```

Question 3

Not answered

Points out of 12.00

A. Apply Holt's linear model to the paperback and hardback book series (data(books), holt()) and compute four-day forecasts in each case.

B. Compare the SSE measures of Holt's method for these two series to those of simple exponential smoothing with ses() method.

C. Compare the forecasts for the two series using both methods. Which is best and why?

```
fcastPaperHolt = holt(books[,"Paperback"], h=4)

plot(fcastPaperHolt)

fcastHardHolt = holt(books[,"Hardcover"], h=4)

plot(fcastPaperHolt)

#--------

SSEPaperOpt = sum(fcastPaperOpt$residuals^2)

SSEPaperHolt = sum(fcastPaperHolt$residuals^2)

SSEHardOpt = sum(fcastHardOpt$residuals^2)

SSEHardHolt = sum(fcastHardHolt$residuals^2)
```

Sums of squared errors are much smaller for Holt's linear method rather than for simple exponential smoothing method in both cases: for Paperback and Hardcover series.

It is clear that Holt's linear method takes into account presence of the upward trend in the data. It leads to a better fit (better one step ahead forecasts).

```
#-------------

par(mfrow=c(1,2)) # This line positions plots side-by-side

plot(fcastPaperOpt, main = "Simple exponential smoothing")

plot(fcastPaperHolt, main = "Holt's linear method")


par(mfrow=c(1,2)) # This line positions plots side-by-side

plot(fcastHardOpt, main = "Simple exponential smoothing")

plot(fcastHardHolt, main = "Holt's linear method")
```

Probably, in both cases, Holt's linear method will predict better than simple exponential smoothing method. It is because Holt's method takes into account clear upward trend in the data, while simple exponential smoothing does not.

Question 4

Not answered

Points out of 12.00

Sam built the forecast model for the quarterly passenger vehicle production using the Holt's method, as depicted in the R snippet code below. After performing decompositional analysis using stl(), he obtained seasonally adjusted data and aplied an additive damped trend Holt's method to this data (damp=TRUE). He used this model to perform a two-year forecast of this quarterly data (h=8) and computed RMSE for one-step forecast. He repeated a similar forecast but without the additive damp trend method as well.

A. You are tasked with using the ets() method for automatically selecting a seasonal model for the data. Which model was selected as the better one?

B. Compare the RMSE of the fitted ets() model with the RMSE of the model that Sam obtained using an STL decomposition with Holt's method. Which gives better in-sample fit (if any)?

C. Compare the two-year forecasts from these two approaches? Which seems more reasonable?

```
require(fpp)
data(ukcars)
plot(ukcars, ylab = "Production, thousands of cars")
stlFit <- stl(ukcars, s.window = "periodic")
plot(stlFit)
adjusted <- seasadj(stlFit)
plot(adjusted)

fcastHoltDamp = holt(adjusted, damped=TRUE, h = 8)
plot(ukcars, xlim = c(1977, 2008))
lines(fcastHoltDamp$mean +
        stlFit$time.series[2:9,"seasonal"],
      col = "red", lwd = 2)

dampHoltRMSE = sqrt(mean(((fcastHoltDamp$fitted + stlFit$time.seri
dampHoltRMSE

fcastHolt = holt(adjusted, h = 8)
plot(ukcars, xlim = c(1997, 2008))
lines(fcastHolt$mean + stlFit$time.series[2:9,"seasonal"],
      col = "red", lwd = 2)

holtRMSE = sqrt(mean(((fcastHolt$fitted + stlFit$time.series[,"sea
holtRMSE
```

etsFit = ets(ukcars)

etsFit

fcastEts = forecast(etsFit, h = 8)

plot(fcastEts)

etsRMSE = sqrt(mean((fcastEts$fitted - ukcars)^2)) etsRMSE

c(HoltRMSE = holtRMSE, EtsRMSE = etsRMSE)

* The forecasts of the Holt's linear trend and ets() methods look very similar and therefore it is very difficult to judge which one is better.

* ets() method has slightly worse fit to the training data than the Holt's linear method. But the forecasts are very similar and it is hard to pick between them.

◄ (DUE: 04/26/2019): SUBMIT: QUIZ: BONUS: TS Forecasting with ARIMA

Jump to...

(DUE: 04/26/2019): SUBMIT: HW: BONUS: TS ARIMA Forecasting ►