ETW3420

Principles of Forecasting and Applications

Topic 3 Exercises

Question 1: Using the %>% (pipe) operator

The pipe operator `%>%` is to help avoid us nesting functions within functions within functions. For example,

`sqrt(mean(tsCV(gogg200, rwf, drift = TRUE, h = 1)^2, na.rm = TRUE))`.

When using the pipe operator `%>%`, the left hand side of each pipe is passed as the first argument to the function on the right hand side. For example, if we type:


```
c(1,2,3) %>% mean()
```

it means that the vector (1,2,3) is passed as the first argument to the `mean()` function.

Otherwise, we would simply just type the command as:

```
mean(c(1,2,3))
```

In the following exercise, we want to do the following:

(i) Produce a seasonal naive forecast for the dataset `a10`.

(ii) Extract the residuals given by $y_t - \hat{y}_t$.

(iii) Produce a plot of the residuals.

Traditionally, one would do the above via the following set of codes:

```
#Produce seasonal naive forecast
fc <- snaive(a10)


#Extract residuals
e <- residuals(fc)


#Plot residuals
autoplot(e)
```


Your turn: Condense the above set of codes into 1 line using the %>% operator.

Question 2

For each of the following series, find an appropriate Box-Cox transformation in order to stabilize the variance. Comment if a Box-Cox transformation was necessary for each of the series.

- usgdp
- mcopper

```
#For usgdp


  #Produce plot of usgdp
autoplot(usgdp)
```

```
  #Calculate BoxCox lambda value
lambda.usgdp <- BoxCox.lambda(usgdp)
print(lambda.usgdp)
```

```
## [1] 0.36635
```

```
  #Plot BoxCox transformed series
    #Method 1
usgdp %>% BoxCox(lambda = lambda.usgdp) %>% autoplot()
```

```
    #Method 2
autoplot(BoxCox(usgdp, lambda = lambda.usgdp))
```


Question 3

Why is a Box-Cox transformation unhelpful for the `cangas` data (monthly Canadian gas production, billions of cubic metres, Jan 1960 – Feb 2005)?

(Hint: What sort of time series are Box Cox transformations designed to handle?)

```
autoplot(cangas)
```

Question 4

(a) Obtain residuals from a seasonal naive forecast applied to the quarterly Australian beer production data (`ausbeer`) from 1992.

You will need to do the following:

- Create a window of data from `ausbeer` that commences from 1992;
- Generate seasonal naive forecasts and plot them;
- Extract residuals and plot them.

(b) Test if the residuals are white noise and normally distributed.

(c) What do you conclude?

Question 5

Consider the sales of new one-family houses in the USA, Jan 1973 – Nov 1995 (data set `hsales`).

(a) Produce some plots of the data in order to become familiar with it.

(b) Split the `hsales` data set into a training set and a test set, where the test set is the last two years of data.

(c) Try using various benchmark methods (i.e. average, naive, seasonal naive, drift) to forecast the training set and compare the results on the test set. Plot the training set, test set and the forecasts. Determine which method did best.

Hint:

1. Use the function `length()` to determine the value of h, the no. of periods in the test set.
2. Use the appropriate functions to produce forecasts by the benchmark methods.
3. Use the `autoplot()` function followed by `autolayers()` to produce the plot. Use the `series` argument in the `autolayers()` function to name your respective forecasts.
4. Use the `accuracy()` function to determine forecast accuracy.

(d) Check the residuals of your preferred method. Do they resemble white noise?

(e) Obtain the RMSE obtained via time series cross-validation.

```
#Obtain forecast residuals

e <- tsCV(hsales, forecastfunction = snaive, h = 4)
print(e)

#Calculate RMSE
```


Question 6

Are the following statements true or false? Explain your answer.

a. Good forecast methods should have normally distributed residuals.

b. The best measure of forecast accuracy is MAPE.

c. If your model doesn't forecast well, you should make it more complicated.

d. Always choose the model with the best forecast accuracy as measured on the test set.