

ETW3420

Principles of Forecasting and Applications

Topic 7 Exercises

Question 1

Daily electricity demand for Victoria, Australia, during 2014 is contained in `elecdaily`. The data is for 365 days. The data for the first 20 days can be obtained as follows.

```
daily20 <- head(elecdaily, 20)
```

- (a) Plot the data and find the regression model for Demand with temperature as an explanatory variable. Why is there a positive relationship?

```
#Time series plot
```

```
autoplot(daily20, facets=TRUE)
```

```
#Scatter plot using ggplot(); rmbr the need to make data a dataframe object
```

```
daily20 %>%
```

```
as.data.frame() %>%
```

```
ggplot(aes(x = Temperature, y = Demand)) +
```

```
geom_point() +
```

```
geom_smooth(method = "lm", se = FALSE)
```

```
#Fit linear regression model
```

```
(fit <- tslm(Demand ~ Temperature, data = daily20))
```

- (b) Produce a residual plot. Is the model adequate? Are there any outliers or influential observations?

```
checkresiduals(fit)
```

- (c) Use the model to forecast the electricity demand that you would expect for the next day if the maximum temperature was 15° and compare it with the forecast if the with maximum temperature was 35°. Do you believe these forecasts?
- To forecast electricity demand, we use the `forecast()` function as usual.
 - However, in this question, we are given specific values of the x-variable (i.e. temperature) for which the forecasts are to be generated.
 - Hence, we need to specify these x-values as a data frame object in the `newdata` argument within the `forecast()` function.

```
fc <- forecast(fit, newdata = data.frame(Temperature = c(15, 35)))  
fc
```

- (d) Give prediction intervals for your forecasts.
- (e) Plot Demand vs Temperature for all of the available data in `elecdaily`. What does this say about your model?

```
elecdaily %>%  
  as.data.frame() %>%  
  ggplot(aes(x = Temperature, y = Demand)) +  
    geom_point() +  
    geom_smooth(method = "lm", se = FALSE)
```

Question 2

The data set `fancy` concerns the monthly sales figures of a shop which opened in January 1987 and sells gifts, souvenirs, and novelties. The shop is situated on the wharf at a beach resort town in Queensland, Australia. The sales volume varies with the seasonal population of tourists. There is a large influx of visitors to the town at Christmas and for the local surfing festival, held every March since 1988. Over time, the shop has expanded its premises,

range of products, and staff.

- (a) Produce a time plot of the data and describe the patterns in the graph. Identify any unusual or unexpected fluctuations in the time series.

```
autoplot(fancy) +  
  xlab("Year") + ylab("Sales")
```

- (b) Explain why it is necessary to take logarithms of these data before fitting a model.

```
# Taking logarithms of the data  
autoplot(log(fancy)) + ylab("log Sales")
```

- (c) Use R to fit a regression model to the logarithms of these sales data with a linear trend, seasonal dummies and a “surfing festival” dummy variable. (Note that in R, the logical arguments TRUE and FALSE take the values of 1 and 0 respectively).

#Execute the following codes and observe what output is generated

```
help(cycle)  
cycle(fancy)  
cycle(fancy) == 3
```

#Create festival dummy:

```
festival <- cycle(fancy) == 3  
festival[3] <- FALSE #i.e. FALSE for March 1987
```

#Print festival dummy

```
festival
```

Fit linear model to logged data (by specifying lambda=0)

```
fit <- tslm(fancy ~ trend + season + festival, lambda = 0)
```

Check fitted values

```
autoplot(fancy) + ylab("Sales") +
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

```
autolayer(fitted(fit), series="Fitted")
```

- (d) Plot the residuals against time and against the fitted values. Do these plots reveal any problems with the model?

```
#Plot residuals against time
```

```
autoplot(residuals(fit))
```

```
#Plot residuals against fitted values
```

```
#Create dataframe first for subsequent ggplot()
```

```
temp.df <- data.frame(Residuals = residuals(fit), Fitted = fitted(fit))
```

```
#ggplot
```

```
ggplot(data = temp.df, aes(x = Fitted, y = Residuals)) +
```

```
geom_point()
```

- (e) What do the values of the coefficients tell you about each variable?

```
coefficients(fit)
```

- (f) What does the Breusch-Godfrey test tell you about your model?

```
checkresiduals(fit)
```

- (g) Regardless of your answers to the above questions, use your regression model to predict the monthly sales for 1994, 1995, and 1996. Produce prediction intervals for each of your forecasts.

```
#Create dataframe of festival for prediction
```

```
future.festival <- rep(FALSE, 36)
```

```
future.festival[c(3, 15, 27)] <- TRUE
```

```
#Produce forecast
```

```
fit %>%
  forecast(h=36, newdata=data.frame(festival = future.festival)) %>%
  autoplot
```

- (h) Transform your predictions and intervals to obtain predictions and intervals for the raw data.
- (i) How could you improve these predictions by modifying the model?

Question 3

The data set `huron` gives the level of Lake Huron in feet from 1875 to 1972.

- (a) Plot the data and comment on its features.

```
autoplot(huron)
```

- (b) Fit a linear regression and compare this to a piecewise linear trend model with a knot at 1915. (Refer to Slide 39 of Topic 7 lecture notes).

```
#Linear Regression
```

```
(fit.lin <- tslm(huron ~ trend))
```

```
#Execute the following codes to observe what outputs they generate
```

```
t <-time(huron)
```

```
t
```

```
help(pmax)
```

```
t - 1915
```

```
pmax(t-1915,0)
```

```
#Piecewise linear trend model
```

```
t <- time(huron)
```

```
tb <- ts(pmax(t-1915, 0))
(fit.pw <- tslm(huron ~ t + tb))
```

- (c) Generate forecasts from these two models for the period up to 1980 and comment on these.

```
#Forecast length
h <- 8

#Forecasts from linear model
fcasts.lin <- forecast(fit.lin, h=h)

#Forecasts from piecewise linear model

#Create the new variables and their corresponding values for forecast period
t.new <- t[length(t)] + seq(h)
tb.new <- tb[length(tb)] + seq(h)

#Combine new variables as data frame
newdata <- cbind(t=t.new, tb=tb.new) %>% as.data.frame()

#Produce the forecast
fcasts.pw <- forecast(fit.pw, newdata = newdata)

#Plot data with predictions and out-of-sample forecasts
autoplot(huron) +
  autolayer(fitted(fit.lin), series = "Linear") +
  autolayer(fitted(fit.pw), series = "Piecewise") +
  autolayer(fcasts.lin, series = "Linear") +
  autolayer(fcasts.pw, series="Piecewise")
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Question 4 (Optional, depending on time)

The `gasoline` series consists of weekly data for supplies of US finished motor gasoline product, from 2 February 1991 to 20 January 2017. The units are in “thousand barrels per day”. Consider only the data to the end of 2004.

- (a) Fit a harmonic regression with trend to the data. Select the appropriate number of Fourier terms to include by minimizing the AICc or CV value.

```
gas <- window(gasoline, end=2004.99)
cv <- numeric(25)
for(k in seq(25))
{
  fit <- tslm(gas ~ trend + fourier(gas, K=k))
  cv[k] <- CV(fit)[‘CV’]
}
k <- which.min(cv)
fit <- tslm(gas ~ trend + fourier(gas, K=k))
```

- (b) Check the residuals of the final model using the `checkresiduals()` function.

```
checkresiduals(fit)
```

- (c) To forecast using harmonic regression, you will need to generate the future values of the Fourier terms. This can be done as follows.

```
fc <- forecast(fit, newdata=fourier(x, K, h))
```

where ``fit`` is the fitted model using ``tslm``, ``K`` is the number of Fourier terms used in creating ``fit``, and ``h`` is the forecast horizon required.

Lets now forecast the next year of data:

```
fc <- forecast(fit, newdata=data.frame(fourier(gas, k, 52)))
```

- (d) Plot the forecasts along with the actual data for 2005. What do you find?

```
autoplot(fc) +  
  autolayer(window(gasoline, start=2005, end=2005.99), series="2005 data")
```

The forecasts look pretty good for the next year.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder