Tutorial 2 Answers

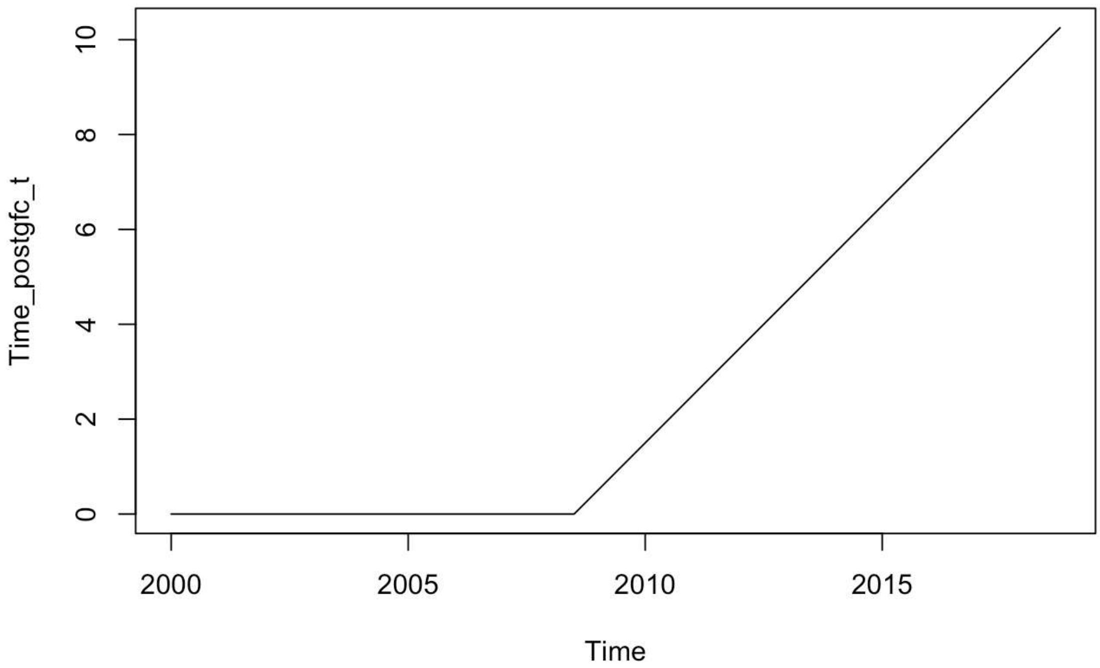
Set up code:

```
dt <- read.csv("RetailSales.csv")
Retail_m <- ts(dt$Total, frequency=12, start=c(1982,4), end=c(2024,9))
Retail_q <- aggregate(Retail_m, nfrequency=4)
log_Retail_q <- log(Retail_q)

Y_t <- window(log_Retail_q, start=c(2000,1), end=c(2018,4))
Time_t <- time(Y_t)
QD_t_ <- factor(cycle(Y_t))
```

1. Below is the code to generate TimePostGFC_t and create the plot.

```
gfc <- 2008.5
Time_postgfc_t <- 1*(Time_t>gfc)*(Time_t-gfc)
plot(Time_postgfc_t)
```



2. Consider the equation

$$Y_t = \beta_0 + \beta_1 \text{Time}_t + \delta \text{TimePostGFC}_t + U_t.$$

a. Until the GFC, i.e. $\text{Time}_t \leq 2008.5$, the definition gives $\text{TimePostGFC}_t = 0$ and hence the equation reduces to

$$Y_t = \beta_0 + \beta_1 \text{Time}_t + U_t.$$

The slope is therefore β_1.

b. After the GFC, i.e. $\text{Time}_t > 2008.5$, the definition gives $\text{TimePostGFC}_t = \text{Time}_t - 2008.5$ and hence the equation becomes

$$\begin{aligned} Y_t &= \beta_0 + \beta_1 \text{Time}_t + \delta(\text{Time}_t - 2008.5) + U_t \\ &= (\beta_0 - \delta \times 2008.5) + (\beta_1 + \delta)\text{Time}_t + U_t. \end{aligned}$$

The slope therefore becomes $\beta_1 + \delta$ after the GFC. If $\delta < 0$ then this would model the slow growth rate after the GFC. Notice this specification also results in a change in the intercept of the regression after the GFC.

c. To check continuity at 2008.5, we can check what happens in the equation as Time_t approaches 2008.5 from below and from above, and verify these two limits are equal. In equation (1):

$$\lim_{\text{Time}_t \uparrow 2008.5} \beta_0 + \beta_1 \text{Time}_t = \beta_0 + \beta_1 \times 2008.5.$$

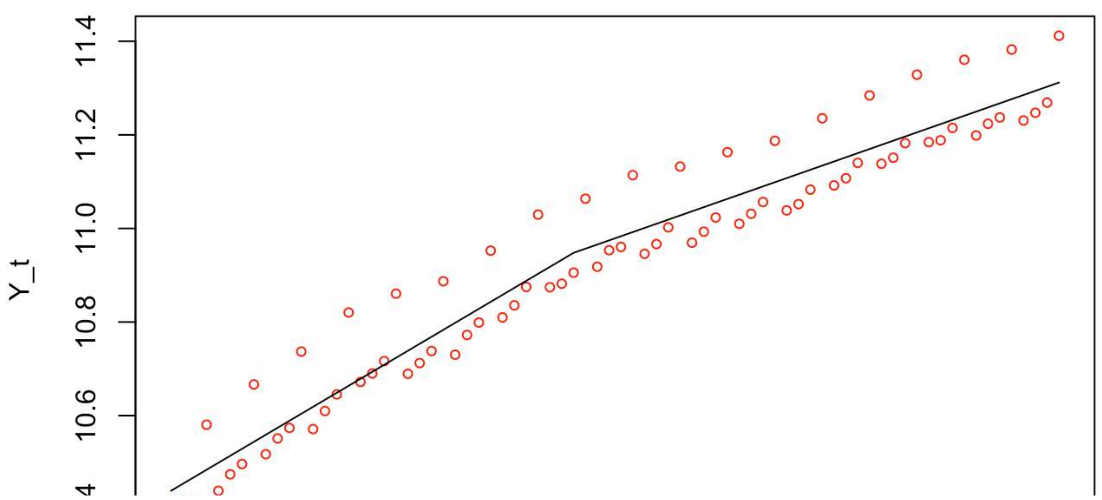
and in (2):

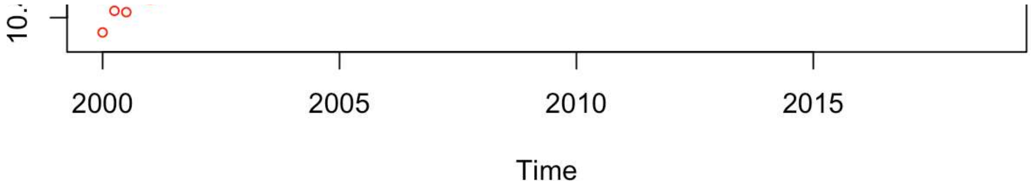
$$\begin{aligned} \lim_{\text{Time}_t \downarrow 2008.5} (\beta_0 - \delta \times 2008.5) + (\beta_1 + \delta)\text{Time}_t &= (\beta_0 - \delta \times 2008.5) + (\beta_1 + \delta) \times \\ &= \beta_0 + \beta_1 \times 2008.5 \end{aligned}$$

as required. Notice the change in the intercept of the regression after the GFC is just right to induce the continuity of the regression at 2008.5.

3. The code to fit the regression of log retail sales on Time_t and TimePostGFC_t is below. It shows the shape of the trend appears to be better modelled by allowing for the trend break at the GFC. Of course the quarterly dummies are not included so the seasonal pattern is not modeled. Since we know that seasonality must be modeled in this time series we wouldn't necessarily need to estimate this particular regression, but the plot without the seasonality provides a clean visualisation of the usefulness of allowing for the trend break.

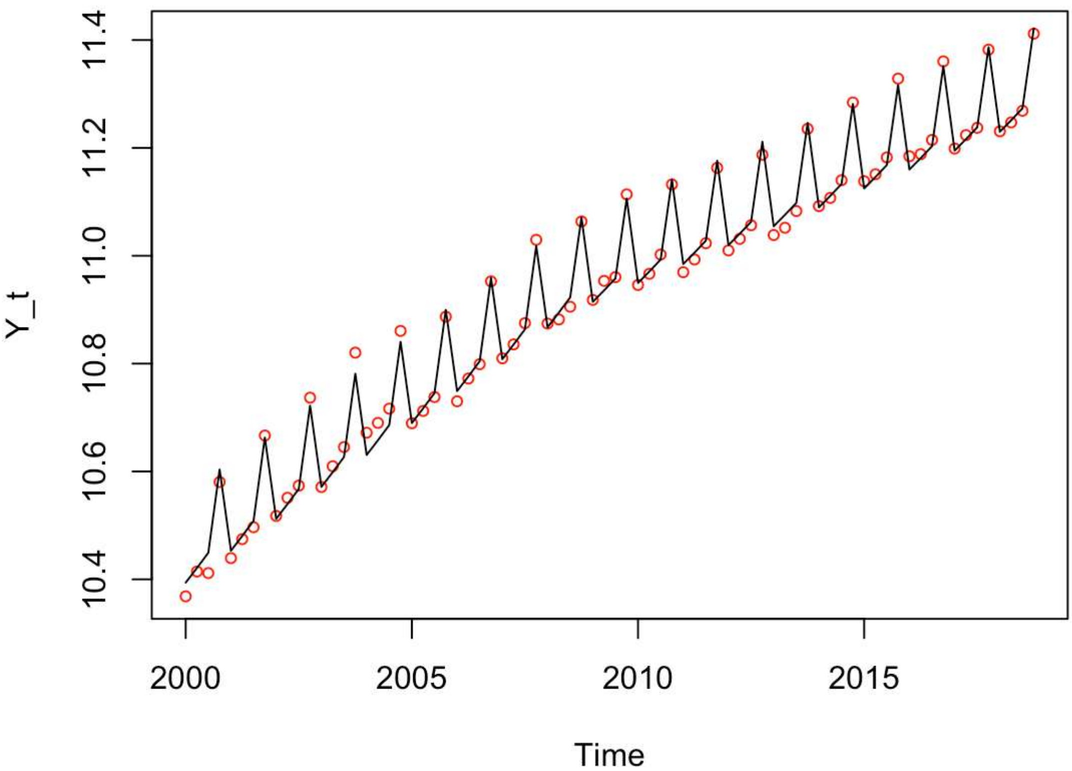
```
# Regression of Y on time trend+GFCtrendbreak
eq3 <- lm(Y_t~Time_t+Time_postgfc_t)
EY_3_t <- ts(predict(eq3), start=c(2000,1), end=c(2018,4), frequency=4)
plot(Y_t, type="p", cex=0.7, col="red")
lines(EY_3_t)
```





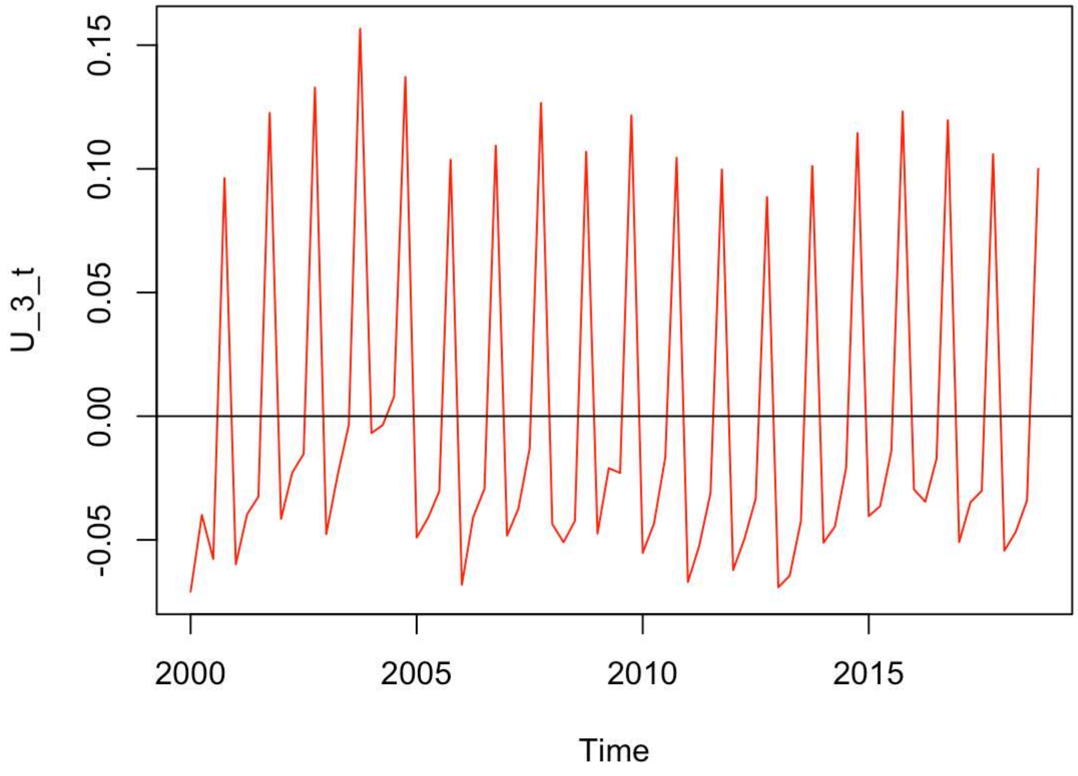
4. Including the quarterly dummies produces a good looking in-sample fit. The overall trend, the change in trend and the seasonality are all modelled. There is nothing visually that is being left unmodelled (although maybe there are still improvements that could be made).

```
# Regression of Y on time trend+QDs+GFCtrendbreak
eq4 <- lm(Y_t~Time_t+QD_t_+Time_postgfc_t)
EY_4_t <- ts(predict(eq4), start=c(2000,1), end=c(2018,4), frequency=4)
plot(Y_t, type="p", cex=0.7, col="red")
lines(EY_4_t)
```



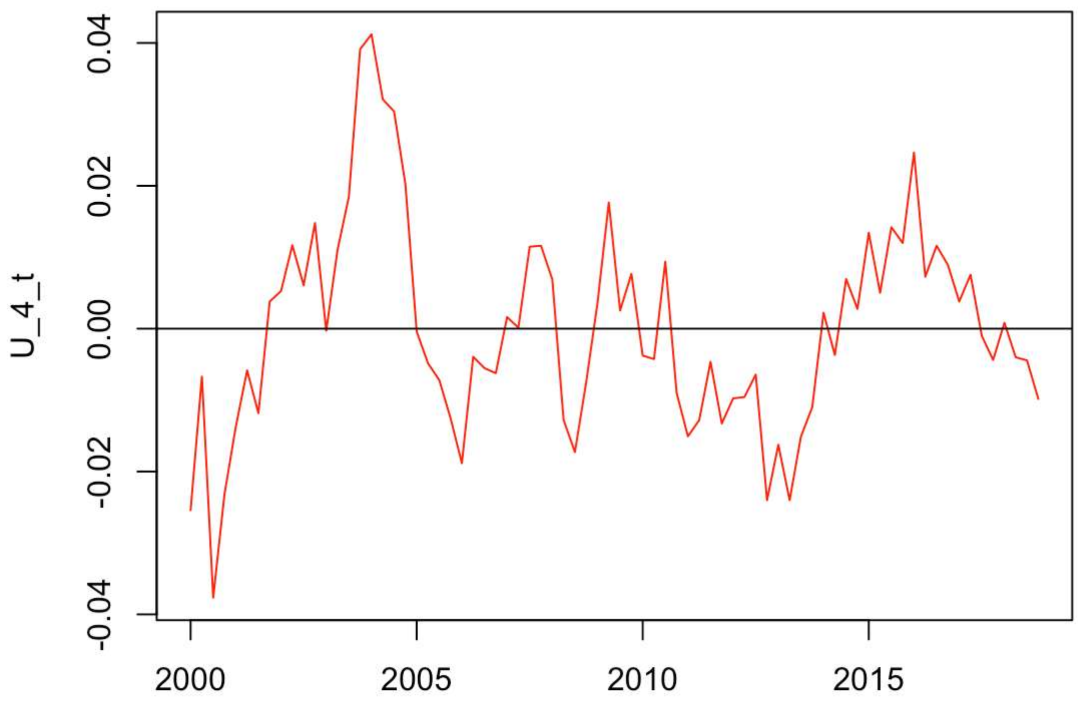
5. The residuals from the regression on linear trend with GFC break are below. The unmodelled seasonality is obvious.

```
# Residuals from regression on time trend+GFCtrendbreak
U_3_t <- ts(eq3$residuals, start=c(2000,1), end=c(2018,4), frequency=4)
plot(U_3_t, col="red")
abline(h=0)
```



Including the quarterly dummies gives the residuals below.

```
# Residuals from regression on time trend+GFCtrendbreak+QDs
U_4_t <- ts(eq4$residuals, start=c(2000,1), end=c(2018,4), frequency=4)
plot(U_4_t, col="red")
abline(h=0)
```



Time

This is the closest yet we have come to apparently ``random'' residuals. However there is still the appearance of some smoothness or cyclical behaviour that may suggest our modeling work here is not yet complete.

6. Forecasts

```
# Actual values of Y
Y_2019 <- window(log_Retail_q, start=c(2019,1), end=c(2019,4))

# Time trend, trend break and seasonal dummies for 2019
Time_2019 <- c(2019, 2019.25, 2019.5, 2019.75)
Time_postgfc_2019 <- 1*(Time_2019>gfc)*(Time_2019-gfc)
QD_2019_ <- factor(cycle(Y_2019))

# Forecast and forecast errors: linear trend+GFCtrendbreak
X3_2019 <- data.frame(Time_t=Time_2019, Time_postgfc_t=Time_postgfc_2019)
EY_3_2019 <- predict(eq3, X3_2019)

# Forecast and forecast errors: linear trend+GFCtrendbreak+QDs
X4_2019 <- data.frame(Time_t=Time_2019, Time_postgfc_t=Time_postgfc_2019,
                  QD_t_=QD_2019_)
EY_4_2019 <- predict(eq4, X4_2019)

Forecasts <- round(cbind(Y_2019, EY_3_2019, EY_4_2019),3)
print(Forecasts)
```

```
        Y_2019 EY_3_2019 EY_4_2019
2019 Q1 11.257    11.321    11.265
2019 Q2 11.277    11.330    11.286
2019 Q3 11.293    11.338    11.308
2019 Q4 11.439    11.347    11.457
```

The regression with GFC trend break and seasonal dummies gives by far the best forecasts of all of the regressions so far. Each forecast is close to the actual value, in each case just slightly too large.

Forecast errors:

```
# Forecast errors: linear trend+GFCtrendbreak
U_3_2019 <- Y_2019-EY_3_2019

# Forecast and forecast errors: linear trend+GFCtrendbreak+QDs
U_4_2019 <- Y_2019-EY_4_2019

ForecastErrors <- round(cbind(U_3_2019, U_4_2019),3)
print(ForecastErrors)
```

```
        U_3_2019 U_4_2019
2019 Q1   -0.063   -0.007
```

```
2019 Q2   -0.053   -0.010
2019 Q3   -0.045   -0.015
2019 Q4    0.092   -0.018
```

The full set of forecast errors confirms they are clearly the smallest for the regression with trend, GFC break and quarterly dummies.