Assignment Task 2

(with Python)

Deadline: 2018-04-06, 23:59

Do not forget to explain your answers!

Note: when generating data use `import numpy as np` and `np.random.seed(student_code)` where `student_code` is your unique student code.

Use the data

```
visitors = [75.7,75.4,83.1,82.9,77.3,105.7,121.9,150.0,
98.0,118.0,129.5,110.6,91.7,94.8,109.5,105.1,95.0,130.3,156.7,190.1,
139.7,147.8,145.2,132.7,120.7,116.5,142.0,140.4,128.0,165.7,183.1,222.8,
161.3,180.4,185.2,160.5,157.1,163.8,203.3,196.9,179.6,207.3,208.0,245.8,
168.9,191.1,180.0,160.1,136.6,142.7,175.4,161.4,149.9,174.1,192.7,247.4,
176.2,192.8,189.1,181.1,149.9,157.3,185.3,178.2,162.7,190.6,198.6,253.1,
177.4,190.6,189.2,168.0,161.4,172.2,208.3,199.3,197.4,216.0,223.9,266.8,
196.1,238.2,217.8,203.8,175.2,176.9,219.3,199.1,190.0,229.3,255.0,302.4,
242.8,245.5,257.9,226.3,213.4,204.6,244.6,239.9,224.0,267.2,285.9,344.0,
250.5,304.3,307.4,255.1,214.9,230.9,282.5,265.4,254.0,301.6,311.0,384.0,
303.8,319.1,313.5,294.2,244.8,261.4,329.7,304.9,268.6,320.7,342.9,422.3,
317.2,392.7,365.6,333.2,261.5,306.9,358.2,329.2,309.2,350.4,375.6,465.2,
342.9,408.0,390.9,325.9,289.1,308.2,397.4,330.4,330.9,366.5,379.5,448.3,
346.2,353.6,338.6,341.1,283.4,304.2,372.3,323.7,323.9,354.8,367.9,457.6,
351.0,398.6,389.0,334.1,298.1,317.1,388.5,355.6,353.1,397.0,416.7,460.8,
360.8,434.6,411.9,405.6,319.3,347.9,429.0,372.9,403.0,426.5,459.9,559.9,
416.6,429.2,428.7,405.4,330.2,370.0,446.9,384.6,366.3,378.5,376.2,523.2,
379.3,437.2,446.5,360.3,329.9,339.4,418.2,371.9,358.6,428.9,437.0,534.0,
396.6,427.5,392.5,321.5,260.9,308.3,415.5,362.2,385.6,435.3,473.3,566.6,
420.2,454.8,432.3,402.8,341.3,367.3,472.0,405.8,395.6,449.9,479.9,593.1,
462.4,501.6,504.7,409.5]
```

The `visitors` dataset - is the monthly Australian short-term overseas visitor data, from May 1985 to April 2005.

Note: the modules `smt.seasonal.seasonal_decompose()`, `sm.graphics.tsa.plot_acf()` and `sm.graphics.tsa.plot_pacf()` **should work** with the specified `visitors` variable type (i.e., a list). In case it doesn't - make sure you are using Python 3 (not 2.7) and make sure that the anaconda installation is not older than a couple of months.

Otherwise, try to specify your series as a `pd.Series()` type object:

```
import pandas as pd

visitors = pd.Series(visitors, name = "Australian Visitors")
new_index = pd.date_range(start = "1985-05", periods = len(visitors.index), freq = "M")
new_index = new_index.to_period()
visitors.index = new_index.to_timestamp()
```

Analyse the data, decompose the deterministic components and forecast the time series:

1. Initial data analysis

1.1 Plot your time series. Does the data look stationary? Why?

1.2 Does the data exhibit autocorrelation? What could be the cause of this type of autocorrelation? (Hint: look at the ACF and PACF - think about what happens to them if a deterministic component is introduced).

1.3 As you know, there are two common decomposition models - additive and multiplicative. Which one would be appropriate for this data and why? (Hint: look at the data variation). If needed, transform your data.

2. Time series decomposition without forecasts

After having an initial look at the data and carrying out any needed data transformations, move on to decompose the deterministic components of your time series. Remember, that one requirement for the seasonal component with period `d` is that $S_t = S_{t+d}$, i.e. the seasonal component itself must not increase as time increases.

2.1 Decompose the data via a moving-average method. Plot the `trend` and `trend + seasonal` components alongside the actual data. Is the decomposed trend and seasonality captured correctly? Explain your answers. (Hint: look at the residual plots as well as their ACF and PACF)

3. Time series decomposition with forecasts

Having an initial idea on the decomposition form and the deterministic components, use a couple of methods for TS decomposition, that allow forecasting:

If needed, generate the `trend` and `seasonal` variables from your data.

3.1 Use the Global Method of OLS to estimate the trend and seasonal components of your data and plot the estimated values from your model. Does the model capture your trend and seasonality components well? Explain why.

3.2 Use the Local Method of Exponential Smoothing (use the double exponential smoothing method) to estimate the deterministic components. Does the model capture the trend and seasonality components well? Explain why.

3.3 Compared with the model from `3.1`, which one is better? Why?

3.4 Forecast your data 12 months ahead using the **best** selected method.