TP 4 – AOS1
Time series analysis

In this practical work session, we will leverage the Python library Pandas to load a time series, model it and make predictions. We will then need the following imports:

```python
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

① Load the time series stored in the file `data/consumption.csv` with `pd.read_csv`. You can use the arguments `parse_dates` and `index_col`. Make sure your dataset is properly loaded by running the following checks :

```python
assert(ts.shape == (164, 1))
assert(type(ts.index) is pd.core.indexes.datetimes.DatetimeIndex)
```

To avoid some future warnings you can also set the frequency of the time series by running

```python
ts.index.freq = "QS"
```

Here the frequency is set to quarter start (`QS`) meaning that the time series is a quarterly one starting at january.

② Plot the time series. Is the time series showing a trend, does it look stationary ?

③ Test the stationarity of the time series by applying an Augmented Dickey-Fuller statistical test (available in the `statsmodels.tsa.stattools` module). What is the p-value ? Is the time series stationary ?

④ Plot the ACF and PACF of the time series. You can use the functions `plot_acf` and `plot_pacf` from the module `statsmodels.graphics.tsaplots`. What kind of models might explain the observed time series ?

⑤ Split the time series into two parts. The first part `ts_train` will be used to train the model and the second part `ts_test` will be used to test it.

To build an ARIMA model, we will use the more general `SARIMAX` class that can also handle seasonality and exogenous variables.

```
from statsmodels.tsa.statespace.sarimax import SARIMAX as sarimax
```

⑥ Fit an ARIMA model on the train set with the parameters identified at question 4.

⑦ Use the `plot_diagnostics` method on the fitted model to display various diagnostics plots. Check that the residuals are normally distributed and not correlated.

Now that the parameters of the model have been learned, we want to make predictions on the test set. To that end, we must first define the same model on the whole time series and give it the learned parameters. This can be done by the following code:

```
# Defining same model on whole time series
model_test = sarimax(ts, order=(p, d, q))

# Use the previouly learned parameters instead of learning them
model_test_res = model_test.filter(model_fit.params)
```

⑧ Use `model_test_res` and its `get_predictions` method with the correct start argument to make one step ahead forecast on the test set.

⑨ Plot the observed test time series along with one step ahead predictions taken from `model_test_res`.

⑩ What is the mean square error of one step ahead predictions on the test set defined by

$$\text{MSE} = \sum_{h=1}^{H} \left(y_{T+h} - \widehat{y}_{T+h:T+h-1} \right)^2.$$

What is the error if we choose another potential model ?

Problem

The dataset in the file `data/debitcards.csv` stores the monthly retail debit card usage in Iceland (million ISK) from january 2000 to december 2012. Give an estimate of the cumulated debit card usage during the 4 first months of 2013.

Particular care should be taken with the frequency of time series, its nature, the choice of the model and its validation.