

Order Prediction with the R language using ERP5

by [OSOE Project](#).

▼ Details

Agenda

- How to Sample Order Data
- How to Extract Order Data
- How to Interface with R
- How to Predict with ARIMA
- How to Display Prediction

▼ Details

A Simple Sales Order

▼ Details

The first step in our experimentation is to create a sample data which will be used to generate the batch of data on which the statistic method will be applied.

So first go to Sales Order Module and fill an order as it is shown in the example above. Keep the order very simple and you must use the product named "Test Product".

Once the order has been created, you must change the workflow status to planned.

Ref: https://www.tiolive.com/G2nV0tfr/sale_order_module/38/

Create Sample Data (Script)

▼ Details

Once the sample order has been created, we can now create the script to duplicate it and modify the quantity according to the cosine function we will use.

The custom script must be created in the 'custom' folder within the 'portal_skins' folder. In order to do so you must go to https://www.tiolive.com/YOURINSTANCEID/portal_skins/custom/manage

Once in this folder you must create a script, 'add' >> 'Script (Python)'. You should choose **SaleOrder_generateCloneData** as script ID. Then click on 'Save & Edit'.

You should then type in the exact same script than above. Since it is a python script, please take care of the case and spaces before the lines.

Once it is saved, you can run the script on the previously created sale order by entering this URL in your browser:

https://www.tiolive.com/YOURINSTANCEID/sale_order_module/NUMBEROFTHEORDER/SaleOrder_generateCloneData

If everything is working properly, it should display "Done"

Ref:

- https://www.tiolive.com/G2nV0tfr/portal_skins/custom/manage

- https://www.tiolive.com/G2nV0tfr/portal_skins/custom/SaleOrder_generateCloneData

- https://www.tiolive.com/G2nV0tfr/sale_order_module/38/SaleOrder_generateCloneData

Watch Sample Data

▼ Details

Then you can go to the Sales Order module and take a look at the generated sale orders.

https://www.tiolive.com/YOURINSTANCEID/sale_order_module

The script might be still running and after some time there should be 101 orders. You can add more if you want to add some randomness in the cosine generation.

Ref: https://www.tiolive.com/G2nV0tfpR/sale_order_module/

Extract Order Data (SQL)

▼ Details

In order to create a time series of orders, we must create the report of the past orders to be used by the method. SO we need to create a new Z SQL Method. You must go back in the '**custom**' folder in '**portal_skins**' then use '**Add**' >> '**Z SQL Method**'. Use '**SaleOrderModule_zGetQuantityList**' as SQL Method ID.

Ref: https://www.tiolive.com/G2nV0tfpR/portal_skins/custom/SaleOrderModule_zGetQuantityList

You can type in the exact same query that is displayed above. This method gets the list of simulation movement with "Test Product Resource", sums their quantity per day. This way, we can know how much was ordered everyday. It is a simple report to create a time series of orders with a day frequency.

Once this Method has been created you can save it.

Test SQL

▼ Details

Once the Method is saved, you should test it thanks to the "Test" tab.

Ref: [https://www.tiolive.com/G2nV0tfpR/portal_skins/custom/SaleOrderModule_zGetQuantityList/manage_test?](https://www.tiolive.com/G2nV0tfpR/portal_skins/custom/SaleOrderModule_zGetQuantityList/manage_test?SUBMIT=Submit+Query)
SUBMIT=Submit+Query

ARIMA (rpy2)

▼ Details

Next step is to create the local python class which will call and use the Arima method using the R language. In order to do so we will create a new Extension in the portal_classes module of ERP5.

First go to portal_classes and add a new extension named '**Rarima**', then edit it.

/YOURINSTANCEID/portal_classes/manage_editExtensionForm?class_id=RArima

Type in the exact same code than the one above. The code imports rpy2, converts a list of floats to a time series using R methods.

It then builds the arima model (fit) and calls the predict function to provide the next **n** value.

Ref: https://www.tiolive.com/G2nV0tfpR/portal_classes/manage_editExtensionForm?class_id=RArima

Creation de classes python locale à une instance. Ça permet de développer des extensions python sur l'instance. Besoin de chercher ce qu'est une classe python.

Add a new Extension called RArima

And edit it

Prediction (External Method)

▼ Details

Now that the extension has been created, we will use it through an External Method. So within the custom folder in portal skin use '**Add**' >> '**External Method**'

Use 'SaleOrderModule_getArimaPredictionList' as external method ID.

You can set the method as it is made above. This Method just invokes the Extension we just created.

Ref: https://www.tiolive.com/G2nV0tfpR/portal_skins/custom/SaleOrderModule_getArimaPredictionList/manage_edit

We are now capable of creating a past series report, and we have a method to call the Arima function using the R language. What we need now is to create the script which will use the method to predict the future based on the past series.

Predict Future Orders (Script)

▢
▼ Details

Create a new script to make a time series consisting of past 10 quantities and future 10 quantities

https://www.tiolive.com/G2nV0tfpR/portal_skins/custom/SaleOrderModule_predictFutureOrderList

This script invokes SaleOrderModule_zGetQuantityList to get past time series.

Then it invokes SaleOrderModule_getArimaPredictionList to the the future.

Then it creates a list of temp objects which we will display later in listbot.

You can play with this script if you want to display results immediately by printing values and display printed values.

Make a New Form

▼ Details

We now create a new form

https://www.tiolive.com/G2nV0tfpR/portal_skins/custom/SaleOrderModule_viewFutureOrderList

With 2 fields: one Listbox and one OOoChart

Form Parameters (Listbox / Chart)

▢
▼ Details

Here are the parameters.

The listbox displays SaleOrderModule_predictFutureOrderList

The chart displays the listbox.

You can first set the listbox parameters and then play with it.

Display Results

▼ Details

Past is displayed in pink, future in green.