Hi Ivan,

Thank you so much for showing interest in our Data Scientist position at Argus Media and for the time for our call today.

We are interested in the way you approach a data science problem and how you present and/or support your reasons with data.

You are asked by the business to provide one day ahead probabilistic forecast (not a point forecast) for the oil prices. The requirement is to use the **gamlss** package in R ( <https://cran.r-project.org/web/packages/gamlss/gamlss.pdf> ) to forecast the probability distribution of the oil prices. The dataset available to you is the “oil” dataset within the **gamlss** package, where the variable “OILPRICE” is the response variable. You are expected to come up with a predictive model on “OILPRICE” and provide the forecast for one day ahead, explaining how you selected the model, the model diagnostics that you used to assess the model etc.

The preferred way to report your findings is an R markdown. Alternative you can report your finding by an R script or any other tool, in a way that does not require further processing.

You can use the code below to install the **gamlss** package in R and extract the oil data.

# install the gamlss package

install.packages(c("gamlss","gamlss.add","gamlss.dist"))

library(gamlss)

library(gamlss.add)

library(gamlss.dist)

# extract the oil dataset

data(oil)

Attached you will find a second proposal test.

You will have 2 weeks (until October 15th), from the receipt confirmation of this email, to provide us with your project.

Any doubt, please let us know.

Best,

**Argus Media – Coding Test**

The purpose of this exercise is to design and implement an entire data preparation pipeline in R. We would like you to implement a robust, extensible and generic framework for data preparation.

Requirements:

1. Take as raw inputs to the data preparation process, the *oil* data from the *gamlss* package.
2. Develop a process that allows us to add additional drivers which are transformations of the raw input timeseries. Include the following transformations:
   1. Rolling standard deviation (of arbitrary window)
   2. Rolling mean (of arbitrary window)
   3. Lagging (of arbitrary order)
   4. Leading (of arbitrary order)
   5. Differencing
   6. Spread (between two input drivers)
   7. Ratio (between two input drivers)
   8. Product (between two input drivers)
3. We must be able to have composition of transformations. Example: First calculate the difference between *OILPRICE* and *resp\_LAG*, and then calculate the rolling standard deviation.
4. The sequence of transformations, and which drivers they act on must be specified by the user. One of the main purposes of this challenge is to develop a generic framework to allow this.
5. For all drivers, either in their raw form or those that results from the application of one or several transformations, we must keep a meta data object where the sequence of transformations is stored. This will allow us to keep track of the meaning of each new driver.
6. For each driver that results from the user-specified sequence of transformations, we need to assess a few statistics:
   1. Normality test
   2. Stationarity test
   3. Correlation coefficient with the target

These statistics need to be stored in the meta data object. The purpose of this is, we may be interested in keeping in the final model only drivers that are normally distributed, or only drivers whose correlation with the target is above a given threshold, or another combination of such criteria.

There is not a single solution to the problem above. The idea of this exercise is to assess the skills of the candidate in designing robust, generic and systematic processes that occur in our data science pipeline.