# Harvard Extension Data Science

# Dynamic Modeling and Forecasting in Big Data

Instructor: William Yu

## Assignment 5

### Dynamic Modeling and Forecasting: Understanding Which Marketing Channel is Effective

- In the TVlift.xlsx, you can see a daily time series data with three marketing spending (sb, snb, tv) and the product sales (sales). See the Description tab for more details. Let's assume that the trainset data is from 4/1/2016 to 2/28/2017 and the testset data is from 3/1/2017 to 3/19/2017.
- Write a script to build some dynamic models based on the trainset data to understand which marketing channel(s) is/are most effective to sell the products. And then use those models to predict the product sales in the testset. Calculate the RMSEs.
  *Note: In this assignment, when we forecast the testset, we are using explanatory variables' actual values in the testset. So this is different from Assignment 4. Why is that? In the real world, you might have some control of your explanatory variables in the real time. A firm might have different plans for its marketing budget. With this model, one can predict what sales would be with different marketing spending.*

- Assignment submission:
  (1) A R script (or Python/Jupyter notebook if you prefer).
  (2) A word/PDF file to show the summary of models and compare their trainset performance and their RMSEs in the testset.

- *Hint:* The following are some examples of the models and a simple way that you can train the model in-sample and forecast out-of-sample (testset) in time series data. I provided 5 models. Feel free to add additional models.

```
full = read_excel('TVlift.xlsx')

train = full[1:334,]

fit01 = lm(sales ~ sb + snb + tv, data=train)
full$predict1 = predict(fit01, newdata=full)
fit02 = lm(sales ~ sb + snb + tv + factor(dow), data=train)   # dow is day of the week seasonality
fit03 = lm(sales ~ lag(sales,1) + sb + snb + tv, data=train)
fit04 = lm(sales ~ lag(sales,1) + sb + snb + tv + factor(dow), data=train)
fit05 = auto.arima(train$sales, xreg = cbind(train$sb, train$snb, train$tv))
test = full[335:353,]
rmse01 = sqrt(mean((test$predict1-test$sales)^2))
```