**Assignment #6**

**Forecasting Fit Metrics**

Due: Wednesday, Oct. 18

(40 pts. Total)

In this assignment we will further expand our exploration of in-sample and out-of-sample (training and testing) metrics. To start we assume we already have narrowed down our choice to five different demand models. So the first task is to load these model into our Global Environment and extract their Information Criteria (IC) metrics. Following are the commands to execute this pre-processing step. A pre-processing R-script is also provided through Canvas. Please study what these commands do one step at a time:

**library**(fpp)

**library**(dplyr)

**load**("model mSD1.rda")  
**load**("model mSD2.rda")  
**load**("model mSD3.rda")  
**load**("model mSD4.rda")  
**load**("model mSD5.rda")  
  
mL <- **list**(mSD1,mSD2,mSD3,mSD4,mSD5)  
mNam <- **c**("mSD1","mSD2","mSD3","mSD4","mSD5")  
cNam <- **c**("CV", "AIC","AICc", "BIC", "AdjR2")  
  
ICSum <- mL %>%   
 **lapply**(CV) %>%   
 **unlist**() %>%   
 **matrix**(byrow=TRUE, ncol=5, dimnames=**list**(mNam,cNam))

ICSum

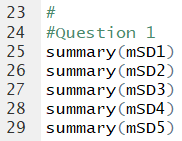
## CV AIC AICc BIC AdjR2  
## mSD1 0.03205782 -1928.863 -1928.711 **-1902.895** 0.9670763  
## mSD2 0.03268963 -1918.550 -1918.398 -1892.582 0.9664643  
## mSD3 0.03374365 -1899.727 -1899.619 -1878.088 0.9652565  
## mSD4 0.03345066 -1904.568 -1904.416 -1878.600 0.9656165  
## mSD5 **0.03182243** -1932.970 **-1932.767** **-1902.674** 0.9673746

We can observe that according to the CV-MSE and AICc the best model is mSD5, while according to the BIC the best model is mSD1 with mSD5 coming a close second.

After examining the ICs, one of your colleagues points out that (1) these ICs reflect the fit of the model over the entire data set and not on any specific store-product combination and (2) there is no reason why you should not apply a different model to each store-product combination. Hence you decide to examine the conventional forecasting fit metrics (ME, MPE, MAE, …, MASE) of each model on each store-product combination and select the best model for each case. The following is a step-by-step process to accomplish this task.

1. (5 pts) Compile a summary of the structure five candidate log-log demand models (i.e., populate the following table with the values of the model’s coefficients).

I ran the summary() function on each model and transferred the data to this table.



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **deal\_X** | **feat\_X** | **deal\_Y** | **LpX** | **H\_Val** |  |
| **mSD1** | 1.37696 | 0.41931 | -0.10771 | -2.25743 | -- | 0.1776 |
| **mSD2** | 1.38368 | 0.41082 | -- | -2.27089 | -- | 0.1792 |
| **mSD3** | 1.36725 | 0.43684 | -- | -2.25630 | -- | 0.1824 |
| **mSD4** | 1.37005 | 0.44088 | -- | -2.24889 | 0.21887 | 0.1815 |
| **mSD5** | 1.37937 | 0.42333 | -0.10590 | -2.25059 | 0.20135 | 0.1768 |

The first building block for our task is to create a function that computes the fit metrics for a given store-product-model combination in-sample or out-of-sample. Following is part of the function specification that you must complete.

*#*  
*# FUNCTION fm(m,SN,PN,SW,EW)*

*# it calculates Fit Metrics of model m*

*#*

*# Inputs:*

*# m - forecasting model (lm-object)*

*# SN - store number (1, ...,7)*

*# PN - product name ("reg" or "lit")*

*# SW - starting week (1 or 41 for in or out of sample)*

*# EW - ending week (40 or 52 for in or out of sample)*

*## Output:*  
*# data.frame object with the fit metrics*  
*# corresponding to 'x'*  
*#*  
fm <- function(m,SN,PN,SW,EW){  
 x <- **select.wsp**(SD,SN,PN,SW,EW)  
 y <- **ts**(**fitted.sales**(x,m), start=SW)

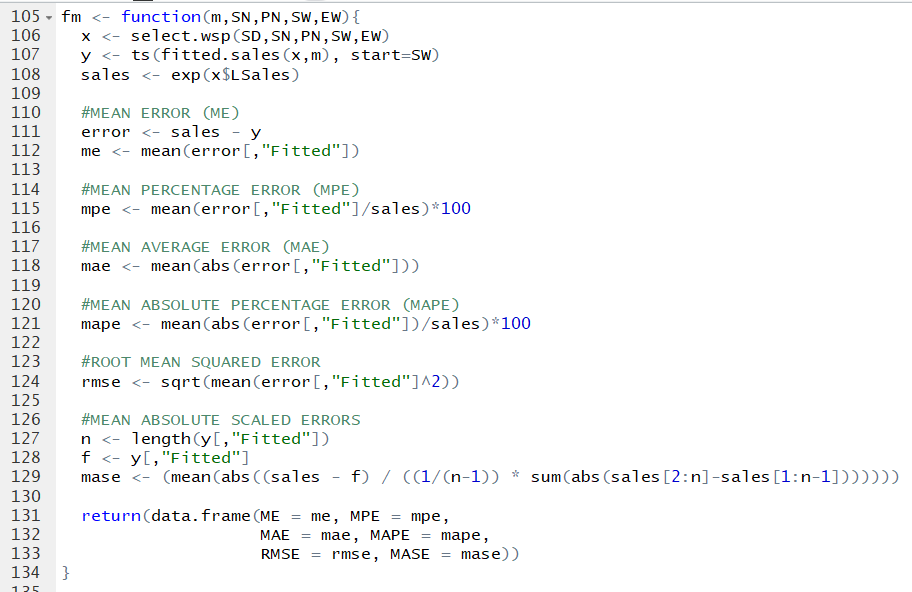
(INSERT NECESSARY CODE HERE)

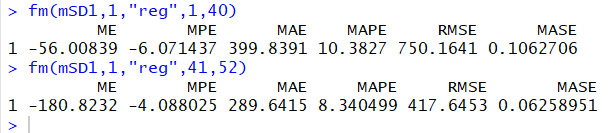
**return**(**data.frame**(ME=me,MPE=mpe,  
 MAE=mae, MAPE=mape,  
 RMSE=rmse, MASE=mase))  
}

Notice that the function above requires the functions select.wsp(…) and fitted.sales(…) that you created in Assignment 4. I am providing my code for these function in the pre-processing R-file.

1. (5 pts) complete the code of the fm(…) function specified above and calculate the in-sample and out-of-sample fit metrics for model mSD1 on store 1 product “reg”. That is, print fm(mSD1,1,”reg”,1,40) and fm(mSD1,1,”reg”,41,52).

Code for the fm() function:





Next we want to apply the fm(…) function over the list of models mL (i.e., use the lapply(…) function) to obtain the fit metric for all the models in the list. To this end you must complete the is.fm(…) function specified below:

*# Function is.fm(sn,pn)*

*# returns block of in-sample fit metrics for*

*# store-product combination*

*#*

*# Inputs:*  
*# sn - store number (1, …,7)*  
*# pn - product name (“reg” or “lit”)*  
*#*   
*# Output:*  
*# Matrix with fit matrix for all*  
*# models in the list for the store*  
*# and product combination*  
*#*  
is.fm <- function(sn,pn){  
 fmNam <- **c**("ME","MPE","MAE","MAPE","RMSE","MASE")  
 FL <- mL %>%

(INSERT NECESSARY CODE HERE)

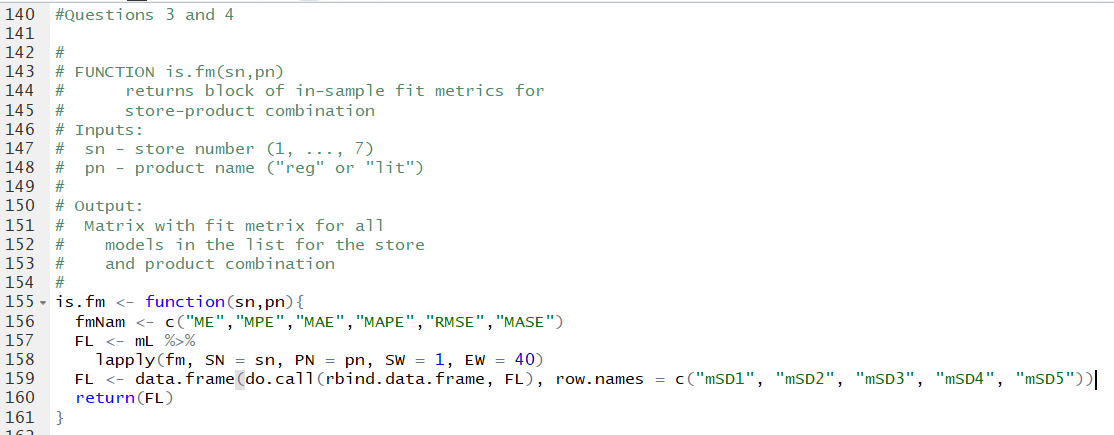
**return**(FL)  
}

When the function is called as is.fm(1,”reg) it should return the following table:

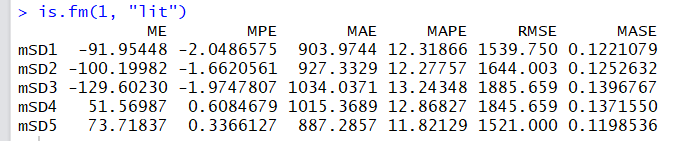
## ME MPE MAE MAPE RMSE MASE  
## mSD1 -56.00839 -6.071437 399.8391 10.382696 750.1641 0.1062706  
## mSD2 -47.50736 -5.850950 402.9136 10.337714 757.3657 0.1070878  
## mSD3 -12.88533 -5.957872 417.8299 10.724873 822.0607 0.1110523  
## mSD4 95.04911 -3.204137 430.2479 10.269286 926.9132 0.1143528  
## mSD5 45.04591 -3.529775 404.6406 9.683872 833.4163 0.1075468

1. (5 pts) Complete the function is.fm(…) specified above and use it to print the fit metrics for store 1 product “lit” .

Code for the is.fm() function:

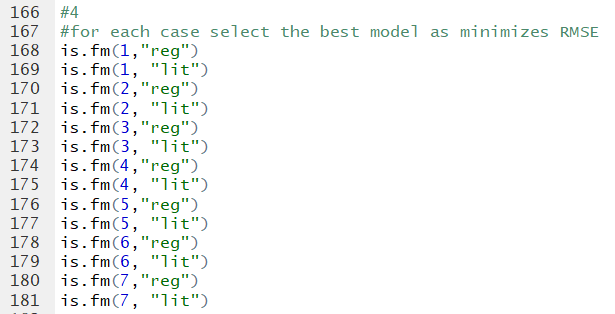


Results for Store 1 product “lit”:



1. (5 pts) Run the function is.fm(…) that you coded in question 3 for each store-product combination and for each case select the best model as the one that minimizes the in-sample RMSE. (i.e., populate the following table with the name of the model selection in each case).

For the results, run lines 168-181:



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Product** | **Store 1** | **Store 2** | **Store 3** | **Store 4** | **Store 5** | **Store 6** | **Store 7** |
| **“reg”** | mSD1 | mSD2 | mSD1 | mSD1 | mSD5 | mSD5 | mSD1 |
| **“lit”** | mSD5 | mSD5 | mSD4 | mSD5 | mSD3 | mSD2 | mSD2 |

Now it is time to compare the in-sample with the out-of-sample performance for the selected models for each store-product combination. To help with this task complete the function os.fm(…) specified below:

*# FUNCTION os.fm(SN,PN,m)*  
*# Calculates in-sample and out-of-sample fit metrics*  
*# for the given stote-product-model combination*  
*#*  
*#Inputs:*  
*# SN - store number (1, ...,7)*  
*# PN - Product name ("reg" or "lit")*  
*# m - modelname (lm object)*  
*#*  
*# Output:*  
*# in-sample and out-of sample statistics*  
*#*  
os.fm <- function(SN,PN,m){  
 rN <- **c**(**paste**("IS",SN,PN,**substitute**(m)),  
 **paste**("OS",SN,PN,**substitute**(m)))  
 OSM <-   
 (INSERT NECESSARY CODE HERE)  
 **rownames**(OSM)<- rN  
 **return**(OSM)  
}

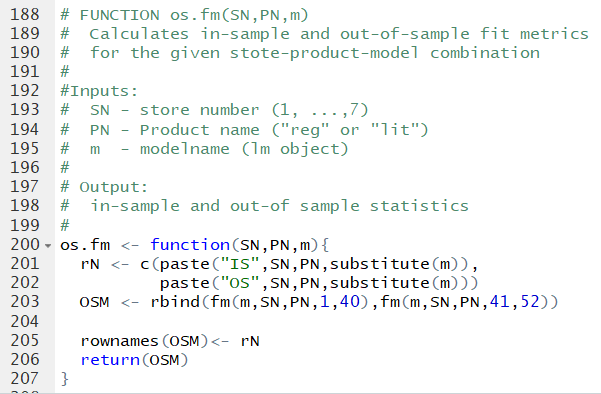
For further reference, when this function is called for store 2 “lit” it should produce the following output:

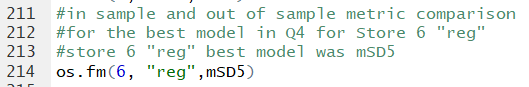
**os.fm**(2,"lit",mSD1)

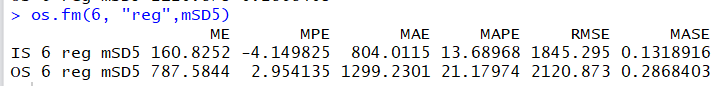
## ME MPE MAE MAPE RMSE MASE  
## IS 2 lit mSD1 -220.5849 -3.610555 1020.638 11.39571 1609.597 0.09005465  
## OS 2 lit mSD1 1429.8138 -4.695595 1854.438 13.84276 5575.006 0.13270110

1. (5 pts) Complete the function os.fm(…) specified above and use it to print the in-sample and out-of-sample metric comparison for the best model you specified in question 4 for store 6 “reg” product.

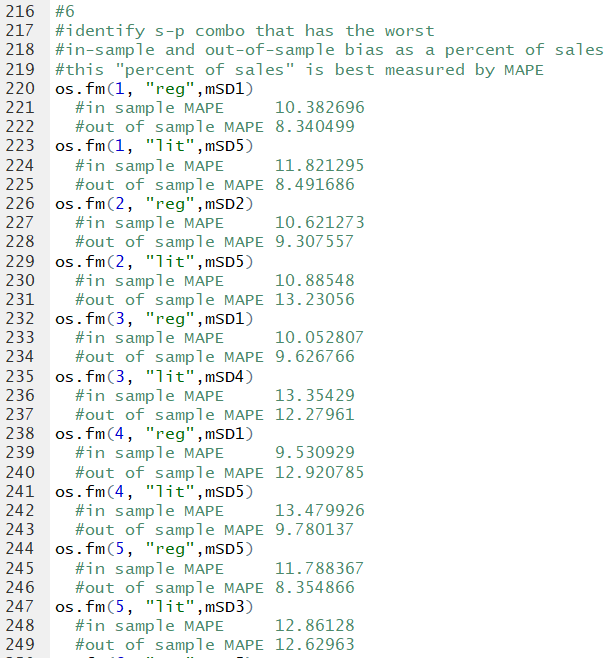
Os.fm function

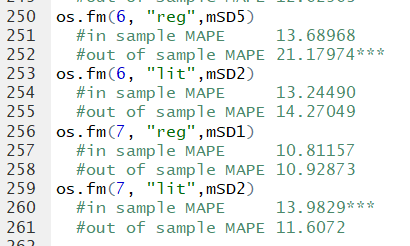






1. (5 pts) Use the function os.fm(…) to identify the store-product combination whose model selection (Question 4) results in the worst in-sample and out-of-sample bias as a percent of sales.





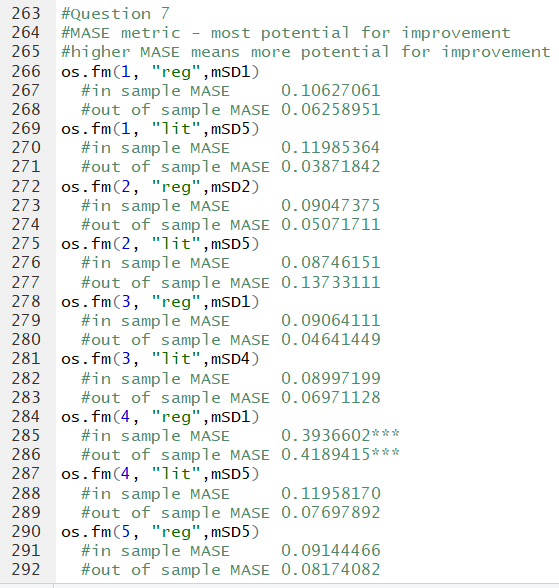
Bias as a percent of sales is best measured by MAPE (Mean Absolute Percentage Error), so we are looking at the highest in-sample and out-of sample MAPE.

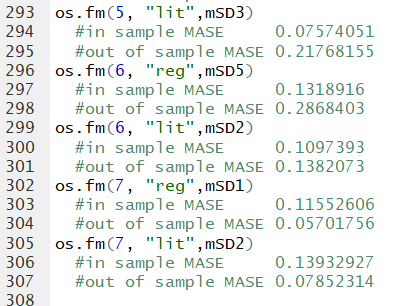
Worst In-Sample Bias: STORE 7, Product “LIT”, with model mSD2 had the worst in-sample bias as a percent of sales (MAPE of 13.9829)

Worst Out-of-Sample Bias: STORE 6, Product “REG”, with model mSD5 had the worst out-of-sample bias as a percent of sales (MAPE of 21.17974)

1. (5 pts) Use the function os.fm(…) and the MASE metric to identify the store-product combination whose model selection (Question 4) has the most potential for improvement.

A higher Mean Absolute Scaled Error means that there is more potential for improvement. We are looking for the highest in-sample and out-of-sample MASE.





Worst In-Sample MASE: STORE 4, Product “REG”, with model mSD1 has the most potential for improvement with a MASE of 0.3936602

Worst Out-of-Sample MASE: STORE 4, Product “REG”, with model mSD1 has the most potential for improvement with a MASE of 0.4189415

1. (5 pts) To complete your report to management, use the spm.fplot(…) function you created in Assignment #5 to include the forecasting plots corresponding to the model-store-product combinations you identified in Questions 6 and 7 above. ( I have included my code for this function in the pre-processing file).

