UNIST
School of Business Administration

FIN552 : High Frequency Financial Data Analysis — Daejin Kim, Spring 2016

Assignment 1
Due Date : **Tuesday, April 4, 2016**

1. **Understanding SAS time informat and format** When you handle data, reading and displaying the are key functions to understand data correctly. In SAS, an "informat" is a specification for how raw data should be read and a "format" is a layout specification for how a variable should be printed or displayed. To understand the format and the informat in SAS system, please read "**SAS_informat_format.pdf**" carefully. Among the first features of SAS that users learn is that SAS dates (and times) have unique characteristics. A SAS date isn't a "standard" numeric or character variable - when displayed it can look like character data but is stored as a number. The date of January 1, 1960, has a value of zero(0): prior dates are negative values, those after have positive values.

   When creating a SAS data set from raw data or from instream datalines, we use an INPUT statement to tell SAS the variable names, types, and lengths. We can also use SAS date INFORMATs to read in dates correctly and ensure that dates are stored correctly as SAS dates. A date INFORMAT tells SAS how to read a value as a SAS date value and store it as an integer which is the number of days since January 1, 1960. An INFORMAT ends with a period, generally specifies the field width, and indicates the form of the value being read.

   Try the following SAS code :

```
/***************/
data one;
input date1 mmddyy10.;
datalines;
4-7-2016
12-4-2015
1-31-2013
;
run;

proc print data = one;
```

```
run;
/***************/
```

How about results? Can you identify the exact date from the printed output? Although the SAS read your data correctly, the printed output cannot be understood by you.

Instead, try printing the data as follows

```
/***************/
proc print data = one;
format date1 date9.;
run;
/***************/
```

In SAS, you can use "PROC CONTENTS" to check the data format. Try the following

```
/***************/
proc contents data = one;
run;
/***************/
```

From this command, you find that the data type of date1 is "Num" and length is "8". How about the format? Now, try the following code. In this code, you need to insert the line "format".

```
/***************/
data one;
input date1 mmddyy10.;
format date1 date9.;
datalines;
4-7-2016
12-4-2015
1-31-2013
;
run;

proc print data = one;
run;

proc contents data= one;
run;
/***************/
```

Do you find differences? In the second experiment, the code in "proc print" didn't include the format statement but correctly displayed. Moreover, if you see the results from the "proc contents", you can find the difference. Right, the format column is added in the output of "proc contents".

You understand the format and informat. Now, read two articles "**SAS_Date.pdf**" and "**SAS_Date2.pdf**" carefully.

The first assignment is reading different types of date variables correctly, create one single final file, and print out the file correctly.

(a) Read the following dates correctly. All dates are same (year = 2005, month = 3, day = 16)

```
03/16/05
03-16-05
03-16-2005
20050316
16-MAR-2005
16MAR05
```

That is, create six (6) separate SAS files (data1, data2, data3, ... , data6) for different types of date informats. For example,

```
data data0;
input date date9.;
format date date9.;
datalines;
16MAR2005
;
```

After reading the above date variables, try this one:

```
data datepractice;
set data0 - data6;
run;

proc print data = datapractice;
run;
```

(b) Submit your sas codes and printed output.