


Attached Files:  [On\\_Time\\_On\\_Time\\_Performance\\_2015\\_6.csv](#) (216.883 MB)  
 [Point\\_of\\_Sale.txt](#) (8.005 KB)

### Problem 1

Point\_of\_Sale.txt is a pipe-delimited text file that contains 100 purchase transactions from a point of sale system. Each transaction contains a variable number of items, and each transaction is written in a single row in the .txt file. The variables in each row of the .txt file are: Transaction ID, Number of Items, Product Code, Units, and Price. Product Code, Units, and Price are repeated in each record <Number of Items> times.

Read the file into a SAS dataset named TRANSACTIONS1, with one observation (row) for each item in each transaction. (Hint: Remember the trailing @ and @@.) The dataset should contain only the following variables:

- Transaction ID,
- a variable named Item\_ID that numbers each item within a transaction from 1 to N,
- Product Code,
- Units,
- Price,
- and a derived variable named Cost that reflects the product of Units and Price.

Create a second SAS dataset named TRANSACTIONS2. TRANSACTIONS2 should contain the same variables and the same rows as TRANSACTIONS1. Additionally, within each Transaction\_ID value create a cumulative sum of COST that increments by the value of COST in each row and resets at each new Transaction\_ID. Use the explicit RETAIN statement to generate these running sums. Also substring the Product Code variable into three new variables Prod1-Prod3, each containing one character from the given Product Code character string. (Hint: use the SUBSTR function).

Submit the following lines of code at the beginning of your SAS program, before any other SAS syntax:

- `title "&SYSUSERID - HW1" ;`
- `ods pdf 'xxxxxx\HW1.pdf';` Replace xxxxxx with a file path on the machine on which the SAS code runs. (This will direct the same output that is rendered by default in an HTML window to a .pdf file as well.

Submit the following line of code at the very end of your SAS program, after all other SAS syntax (to close the ods pdf statement and generate the actual pdf file):

- `ods pdf close ;`

Use PROC PRINT to print each dataset (TRANSACTIONS1 and TRANSACTIONS2).

Use PROC FREQ to generate the distribution of values of variables Prod1-Prod3. Use PROC FREQ to generate the two-way distribution of Prod2 and Prod3.

Use PROC MEANS to generate summary statistics MIN, MAX, MEAN, and STDEV for each numeric variable in the Transactions dataset. (Hint: Use the keyword `_NUMERIC_` in the VAR statement to reference all numeric variables.)

Use PROC MEANS to generate summary statistics MIN, MAX, and MEAN for each numeric variable for each Transaction ID.

Use PROC UNIVARIATE to generate descriptive output for the variable COST. Include appropriate syntax to generate a histogram of the values of COST, and overlay both a normal and a lognormal distribution over the histogram.

## Problem 2

Read the BTS monthly flight on-time performance file for June 2015 into a SAS dataset named BTS201506. Use the code that has been reviewed in class, including the statements that create new variables for the one-period lag delays and lag delay indicators.

Create two new variables, named DepDelayLag2 and ArrDelayLag2, that reflect the two-period departure and arrival lags for each flight. Ensure that the new lag(2) variables have a value of zero for the first and second flights of the day for any given aircraft.

Use PROC UNIVARIATE to generate a description of the variables ArrDelay, ArrDelayLag, and ArrDelayLag2. Include a histogram of each with a normal and lognormal distribution overlay. (Note: Reference the SAS documentation to include THETA= for the lognormal overlay.)

Use PROC CORR to generate a correlation matrix for the variables DepDelay, ArrDelay, ArrDelayLag, and ArrDelayLag2. Use appropriate ODS statements to direct PROC CORR output to a SAS dataset named BTSCorr. Print the contents of BTSCorr with PROC PRINT ;

Use all non-cancelled flights in BTS201506 to estimate a regular OLS regression model with DepDelay as the DV and the following as the IVs:

- CRSDepTime
- seqnum
- DepDelayLagInd
- DepDelayLag
- DepDelayLagCum
- ArrDelayLagInd
- ArrDelayLag
- ArrDelayLagCum
- DepDelayLag2
- ArrDelayLag2

Add appropriate syntax to add the Variance Inflation Factor to the output.

Next, use all non-cancelled flights in BTS201506 to estimate a separate LOGISTIC regression model for each carrier. Create a new variable named DepDelayIND, defined as 1

when DepDelay is greater than 15, and 0 otherwise. Specify DepDelayInd as the response variable and use the same IVs as in the OLS regression model above.