NOTE #6: DESCRIPTIVE AND UNIVARIATE STATISTICS I

PROC MEANS;

```
PROC MEANS <DATA=mydata>  <list of statistics> <options>;
  VAR variable1 variable2, …;
  OUTPUT OUT=outdataname
         Statistics = variables;
RUN;
```

Mostly used Statistics in PROC MEANS:

CLM	Lower and Upper 95% confidence interval for mean
LCLM/UCLM	95% Lower/Upper Confidence Limit for mean
KURT	Kurtosis
MAX	Maximum
MEAN	Average
MEDIAN	Median
MIN	Minimum
N	Number of observations without missing
NMISS	Number of observations with missings
PROBT	Probability of a greater absolute value for t-value
P95	95th percentile (also available P1, P5, P10, P25, P50,P75, P90, P99)
Q1 / Q3	25th / 75th percentile
RANGE	Range
STD	Standard deviation
SUM	Sum
T	t-test for Ho: mean = 0
VAR	Variance

```
/* Example 5-1 */

DATA Ex5_1;
INPUT Class $ Gender $ Score;
DATALINES;
A Male 96
A Male 87
A Male 89
A Female 98
A Female 82
B Male 65
B Male 85
B Female 63
B Female 93
B Female 77
C Male 62
C Male 94
C Male 80
C Female 99
;
PROC MEANS DATA = ex5_1 ;
BY Class; RUN;  * To use BY statement data must be sorted by the variable;

PROC SORT DATA = ex5_1 OUT=ex5_1sort;
by Gender; run;
```

```
PROC MEANS DATA = ex5_1sort MEAN N;
 BY Gender; RUN;

PROC MEANS DATA = ex5_1 chartype; * this option will give you binary _TYPE_;
 Class Class Gender; *Class statement doesn't require sorted data;
 OUTPUT OUT = ex5_1out
  N = count
  Mean = meanscore;
RUN;
Proc Print DATA=ex5_1out; run;
Proc Print DATA=ex5_1out (DROP = _FREQ_) ;
 Where _TYPE_ EQ '11'; *note _TYPE_ is a character variable;
RUN;


/* Example 5-2 */

DATA GNP; SET SASHELP.GNP;
Year =year(date);
quarter = qtr(date);

PROC MEANS DATA = GNP chartype;
VAR GNP CONSUMP INVEST EXPORTS GOVT;
CLASS quarter;
OUTPUT OUT = gnp_out
       /* (drop = _:) will remove all variable beginning with an underscore */
   N (quarter) = count
   MEAN (GNP CONSUMP)=
   STD (GNP CONSUMP) =
   MAX (INVEST EXPORTS) =
   LCLM (GNP) = UCLM (GNP)= / autoname;
run;

PROC PRINT data = gnp_out heading=horizontal; RUN;

DATA gnp_CI; SET gnp_out (KEEP=Quarter GNP_Mean GNP_StdDev count);
  DO i =1 to 5;
   IF _N_=i then
    DO;
      LL = GNP_Mean - TINV (.975, count-1) * GNP_StdDev/SQRT(count);
      UL = GNP_Mean + TINV (.975, count-1) * GNP_StdDev/SQRT(count);
     END;
  END;
  DROP i;
RUN;

Proc Print DATA = gnp_CI heading=horizontal; run;


/* Example 5-3 */
/* This example simulate 100 random samples of each size 30 from N(MU, STD) and
calculate 95% CI for mean for each sample */

DATA Rand_Norm;
Count=100; N=30; MU=5; STD=2; seed=0;
  DO I=1 TO Count;
       DO K=1 TO N;
       X=MU+STD*RANNOR(seed);  OUTPUT;
      END;
  END;
RUN;

PROC PRINT; RUN;
```

```
PROC MEANS NOPRINT MEAN; BY I; VAR X;
 OUTPUT OUT=mean_out MEAN=Mean STD= SD  N= N ;
RUN;

DATA CI; SET mean_out; MU=5;
 LOWER=Mean-TINV(.975, N-1)*SD/SQRT(N);
 UPPER=Mean+TINV(.975, N-1)*SD/SQRT(N);

 IN=0;  IF LOWER<MU<UPPER THEN IN=1;

PROC MEANS MEAN SUM;
 VAR Mean SD LOWER UPPER IN;
RUN;

/* IN-CLASS

Generate 100 random samples from Poisson distribution with a mean of your
choice and calculate 90% CI (a) using normal approximation (2) using 5% and 95%
tiles. Count the number of intervals which contain the true mean and compare.

*/
```

Functions for Random Samples:

Distribution	SAS Function
Binomial (n,p)	RANBIN(seed,n,p)
Exponential (lambda)	ranexp(seed)/lambda
Beta (alpha, beta)	beta*rangam(seed,alpha);
Normal (mu, sigma)	mu+sigma*rannor(seed);
Poisson (mean)	RANPOI(seed,mean)
Uniform (b, a+b)	a*ranuni(seed)+b

PROC FREQ; The procedure provides tables (one, two, and three ways) of counting frequencies of both character and numeric variables.

```
/* Example */
/* Consider the data in Example 5-1 */

Proc Format;
 value grade 0 -< 70 = 'C to D'
             70 -< 90 = 'B to C'
             90 - HIGH = 'A to B';
RUN;

PROC FREQ DATA=Ex5_1 order=formatted;
                *also available order=data, order=freq;
 Format Score grade.;
                *format is used to convert numeric to character category;
 Tables Class / nocum nopercent;
                *One-way table. nocum removes cumulative statistics columns;
                *nopercent will not give percent;
 Tables Gender*Score / Chisq;
                *Chisq gives test for independence between gender and score;

 Table Score * (Class Gender) / nocol norow fisher;
                    *two two-way tables: Score*Class, Score*Gender;
                    *nocol norow will remove conditional prob and ;
```

```
                    *fisher also gives Fisher's Exact test for independence;
RUN;
```

PROC TABULATE : This procedure can be used to generate tabular reports which involve descriptive statistics.

```
PROC TABULATE DATA = Ex5_1;
 CLASS Gender Class;
 var Score;
 table (Class ALL)*(N*f=5.0 PCTN);
 table Score*mean*f=7.1;
 table (Gender ALL)*(Class ALL), Score*(N mean min max)*f=6.2;
 Keylabel ALL = 'Overall'
          MIN = 'Lowest'
          MAX = 'Highest'
          PCTN = 'Percent';
 run;
```