**BA EXPERIMENT 3**

**DESCRIBE MISSING DATA:**

**CODE:**

ods noproctitle;

proc format;

value \_nmissprint low-high="Non-missing";

value $\_cmissprint " "=" " other="Non-missing";

run;

proc freq data=WORK.IMPORT;

title3 "Missing Data Frequencies";

title4 h=2 "Legend: ., A, B, etc = Missing";

format 'RAM (GB)'n 'Screen Size (inch)'n 'Battery Life (hours)'n

'Weight (kg)'n 'Price ($)'n \_nmissprint.;

format Brand Processor Storage GPU Resolution

'Operating System'n $\_cmissprint.;

tables Brand 'RAM (GB)'n Processor Storage GPU 'Screen Size (inch)'n

Resolution 'Battery Life (hours)'n 'Operating System'n 'Weight (kg)'n

'Price ($)'n / missing nocum;

run;

proc freq data=WORK.IMPORT noprint;

table Brand \* 'RAM (GB)'n \* Processor \* Storage \* GPU \* 'Screen Size (inch)'n

\* Resolution \* 'Battery Life (hours)'n \* 'Operating System'n \* 'Weight (kg)'n

\* 'Price ($)'n / missing out=Work.\_MissingData\_;

format 'RAM (GB)'n 'Screen Size (inch)'n 'Battery Life (hours)'n

'Weight (kg)'n 'Price ($)'n \_nmissprint.;

format Brand Processor Storage GPU Resolution

'Operating System'n $\_cmissprint.;

run;

proc print data=Work.\_MissingData\_ noobs label;

title3 "Missing Data Patterns across Variables";

title4 h=2 "Legend: ., A, B, etc = Missing";

format 'RAM (GB)'n 'Screen Size (inch)'n 'Battery Life (hours)'n

'Weight (kg)'n 'Price ($)'n \_nmissprint.;

format Brand Processor Storage GPU Resolution

'Operating System'n $\_cmissprint.;

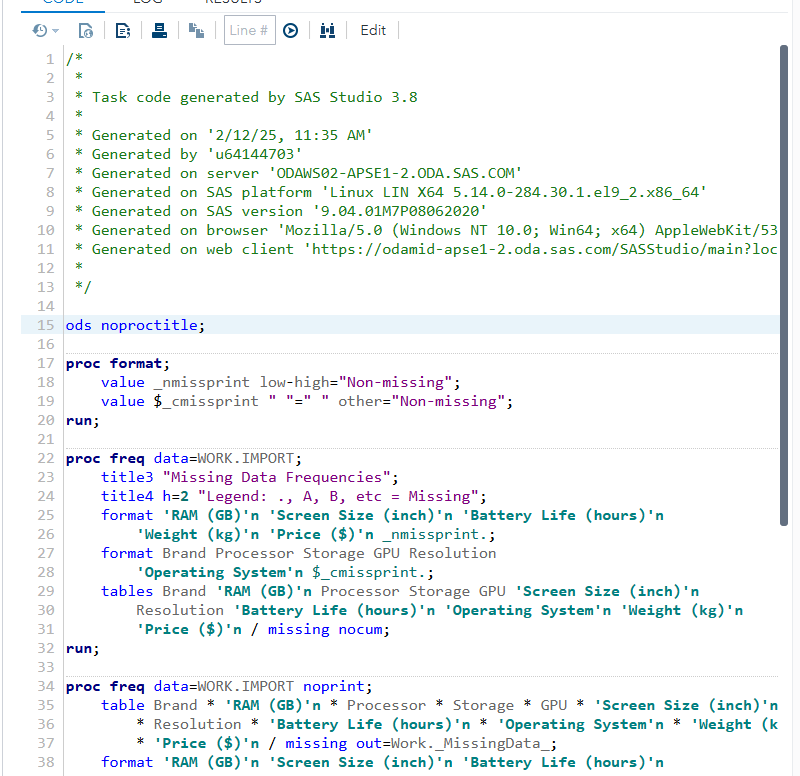
label count="Frequency" percent="Percent";

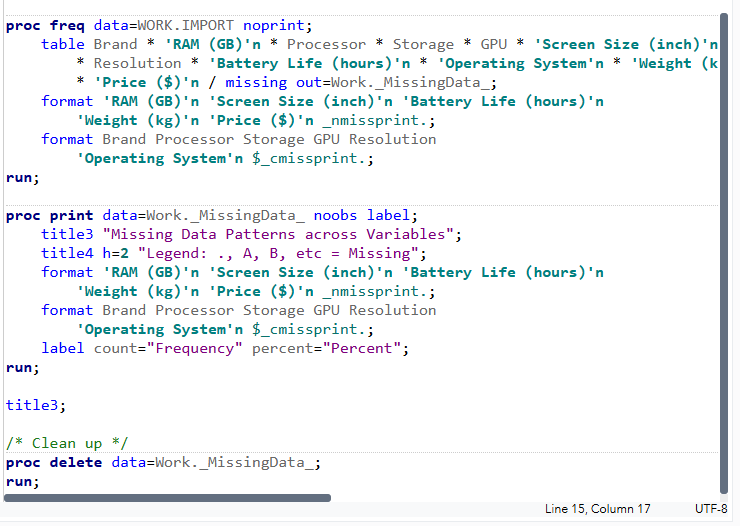
run;

title3;

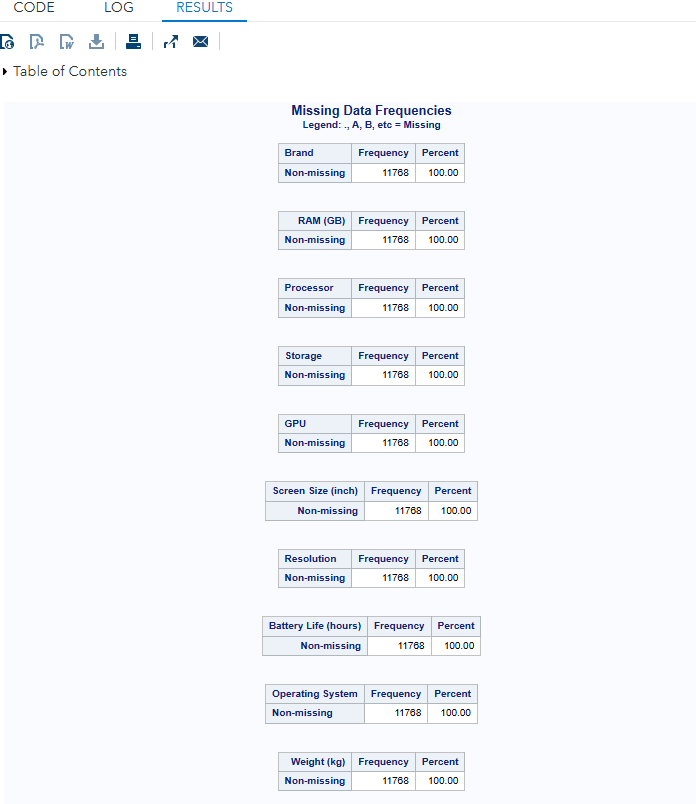
proc delete data=Work.\_MissingData\_;

run;





**RESULTS:**





**STANDARIZE DATA:**

**CODE:**

ods noproctitle;

proc stdize data=WORK.IMPORT method=std nomiss out=work.Stdize oprefix

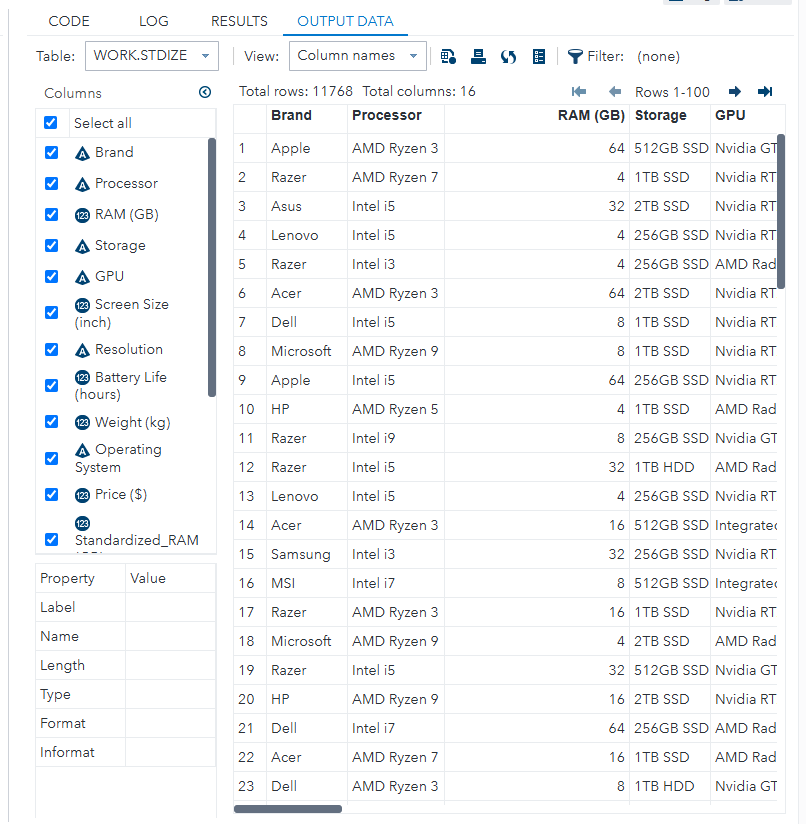
sprefix=Standardized\_;

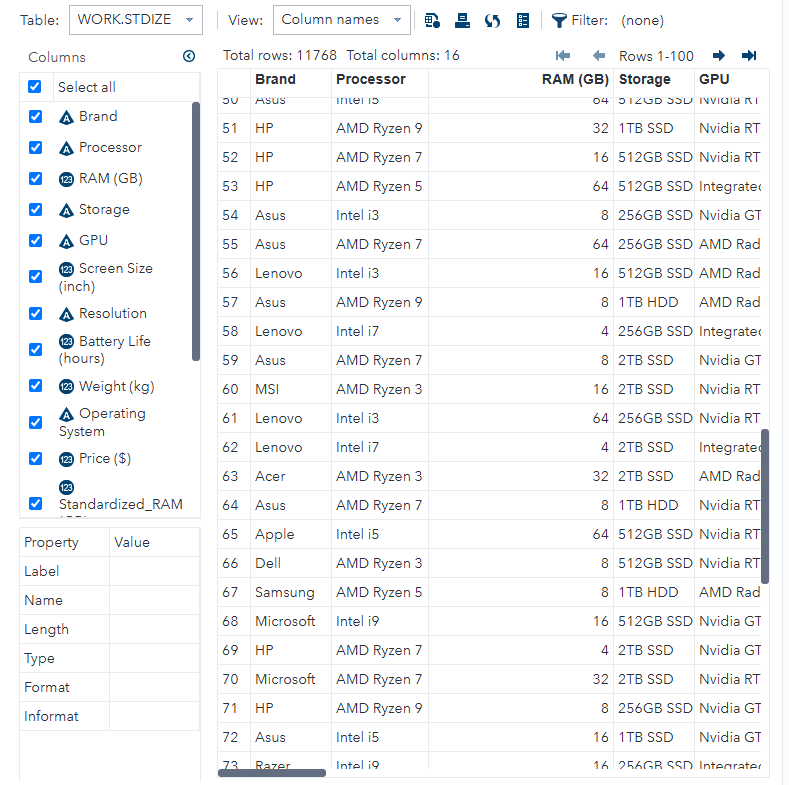
var 'RAM (GB)'n 'Screen Size (inch)'n 'Battery Life (hours)'n 'Weight (kg)'n

'Price ($)'n;

run;

**OUTPUT DATA:**





**FILTER DATA:**

**INPUT:**

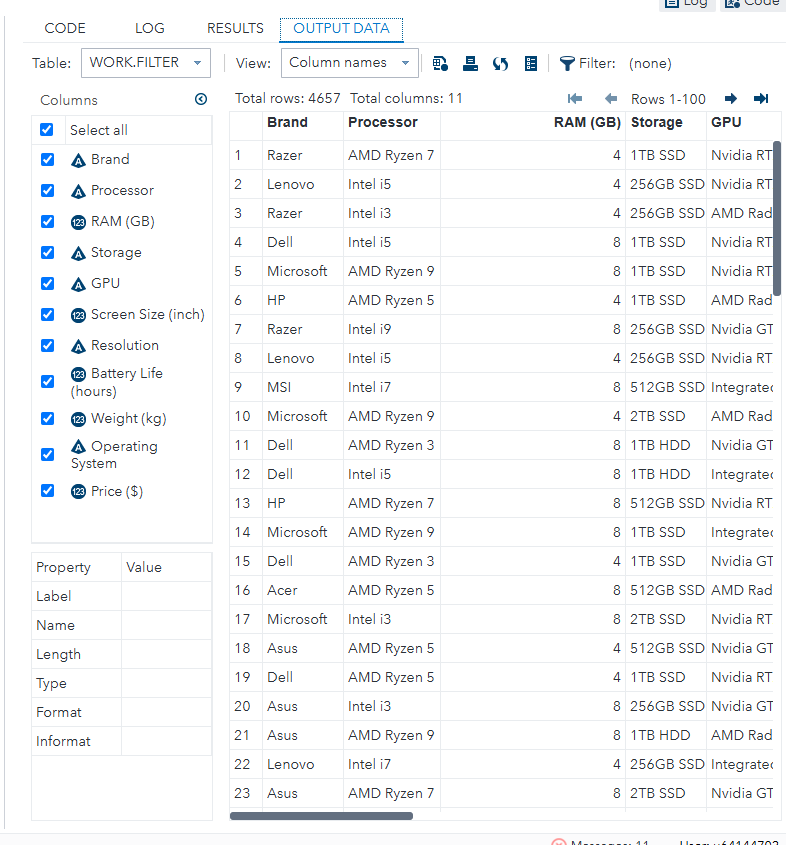
proc sql noprint;

create table work.filter as select \* from WORK.IMPORT where('RAM (GB)'n LT

input('16', BEST12.) );

quit;

**OUTPUT DATA:**



**GRAPH:**

**BAR CHART:**

INPUT:

ods graphics / reset width=6.4in height=4.8in imagemap;

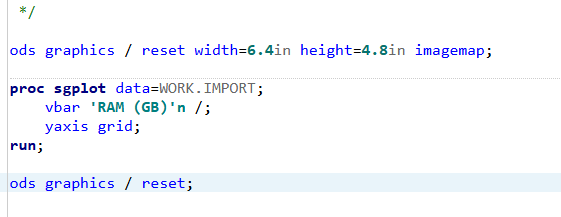
proc sgplot data=WORK.IMPORT;

vbar 'RAM (GB)'n /;

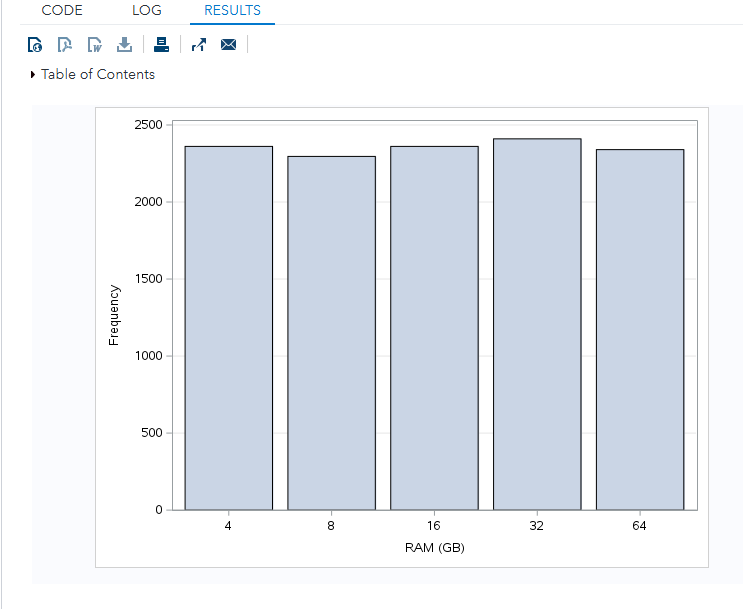
yaxis grid;

run;

ods graphics / reset;



RESULTS:



**BUBBLE PLOTS:**

**CODE:**

ods graphics / reset width=6.4in height=4.8in imagemap;

proc sgplot data=WORK.IMPORT;

bubble x='RAM (GB)'n y='Screen Size (inch)'n size='Weight (kg)'n/

group=Storage bradiusmin=7 bradiusmax=14;

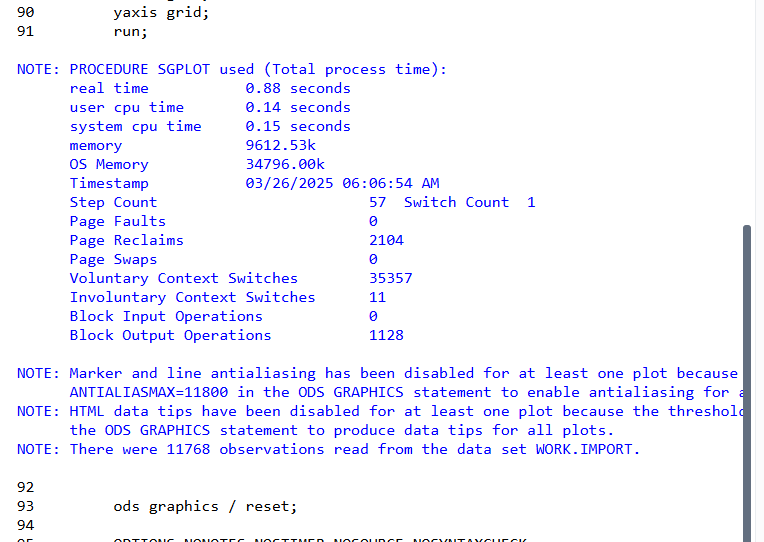
xaxis grid;

yaxis grid;

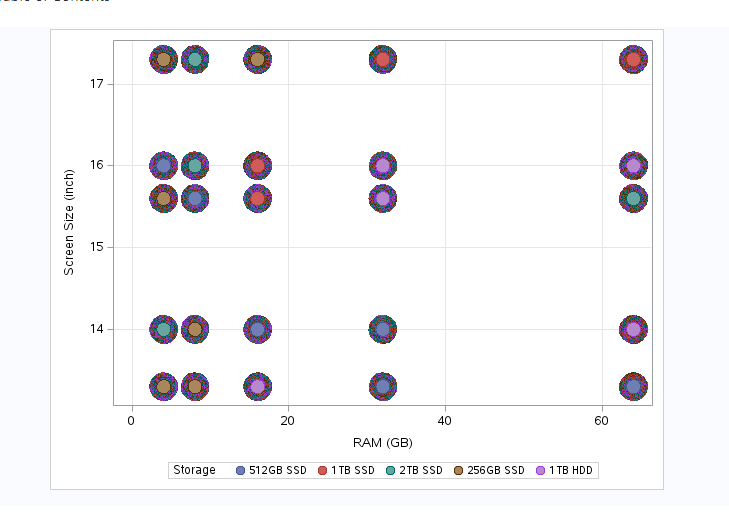
run;

ods graphics / reset;

**LOG:**

****

**OUTPUT:**

****

**HISTOGRAM:**

**CODE:**

ods graphics / reset width=6.4in height=4.8in imagemap;

proc sgplot data=WORK.IMPORT;

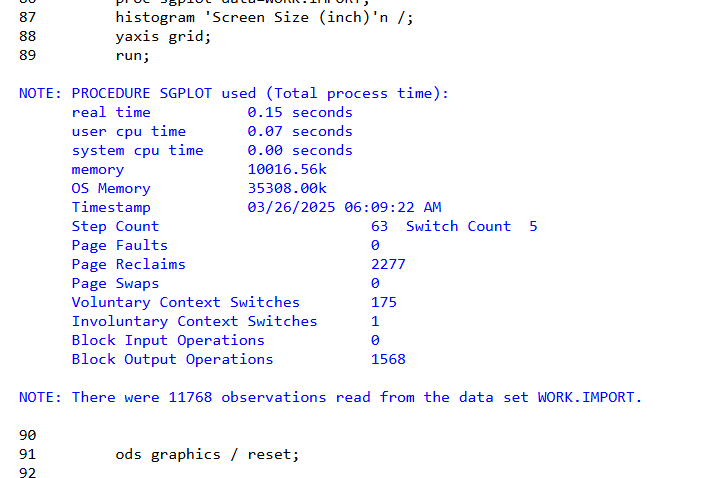
histogram 'Screen Size (inch)'n /;

yaxis grid;

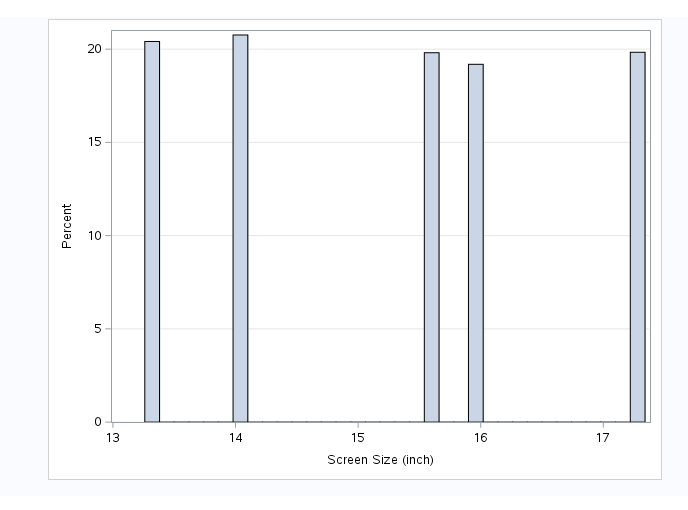
run;

ods graphics / reset;

**LOG:**

****

**OUTPUT:**

****

**SUMMARY OF STATISTICAL DATA:**

**ONE WAY FREQUENCY:**

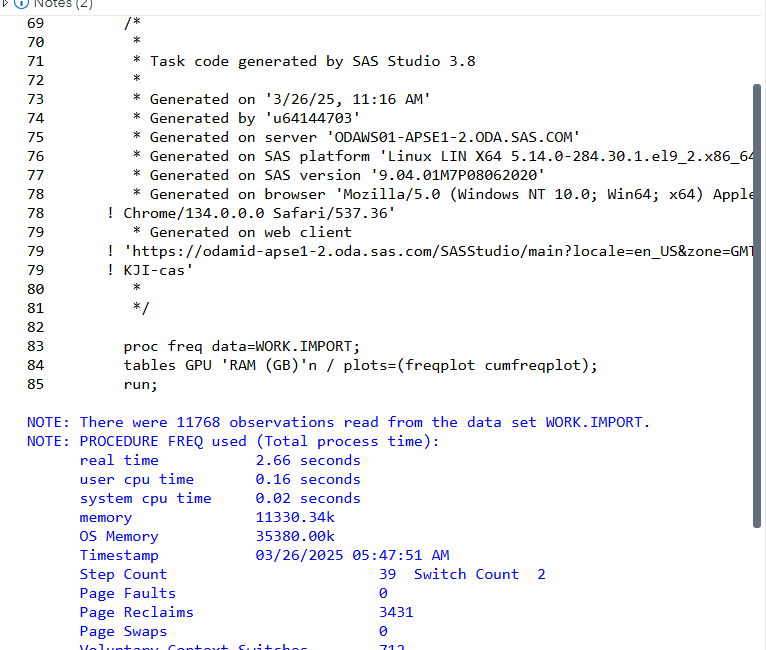
**CODE:**

proc freq data=WORK.IMPORT;

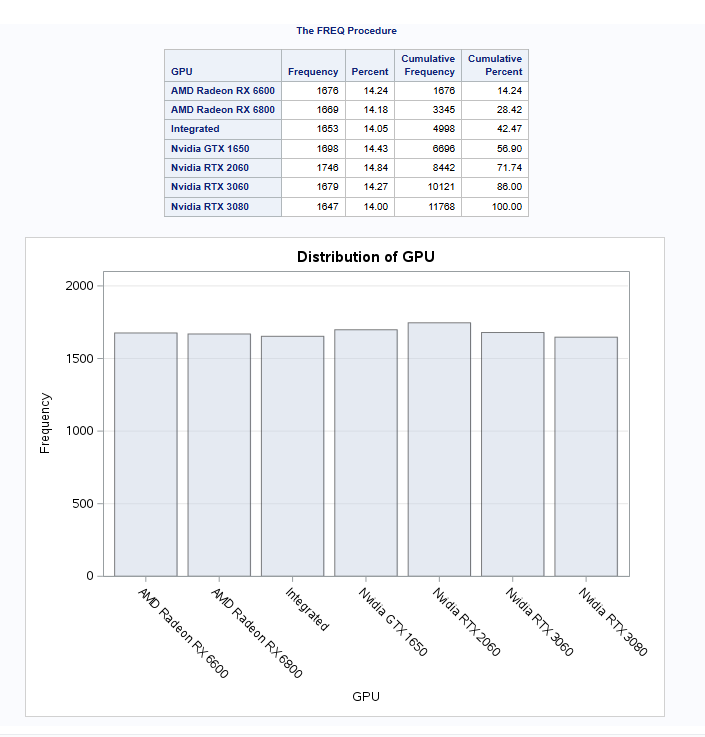
tables GPU 'RAM (GB)'n / plots=(freqplot cumfreqplot);

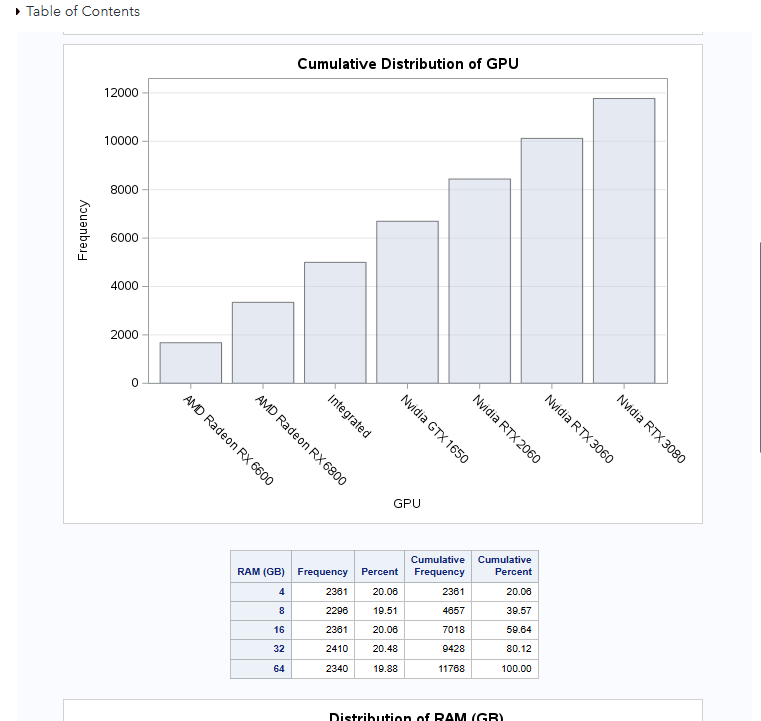
run;

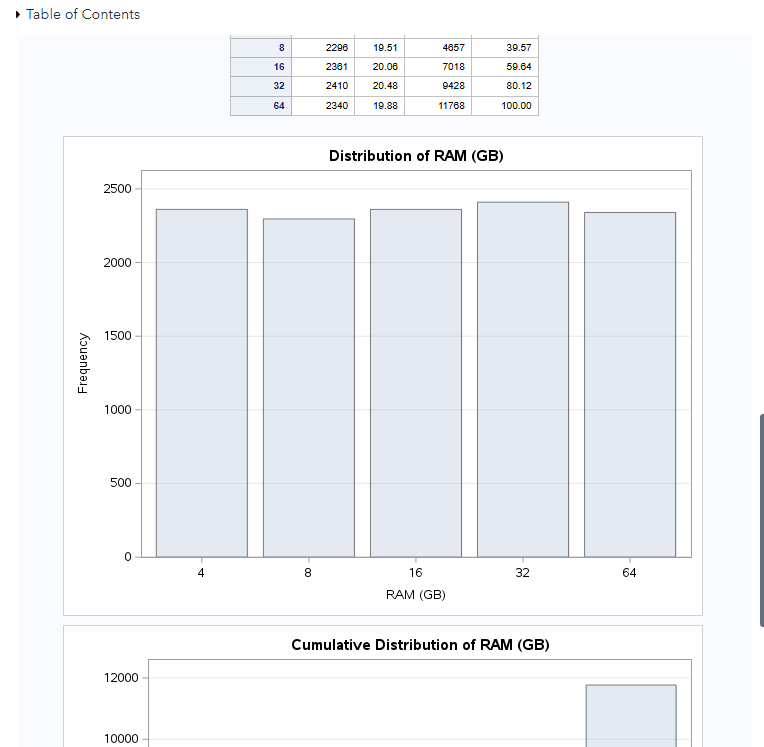
**LOG:**

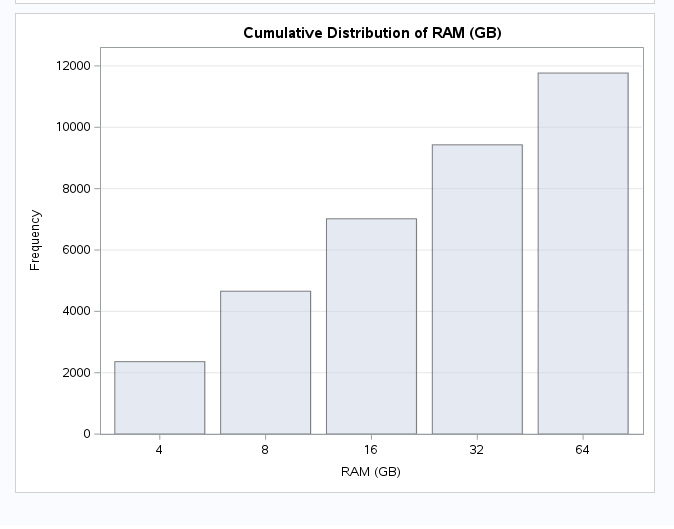
****

**OUTPUT:**

****

****

****

****

**PROC MEANS:**

**CODE:**

ods noproctitle;

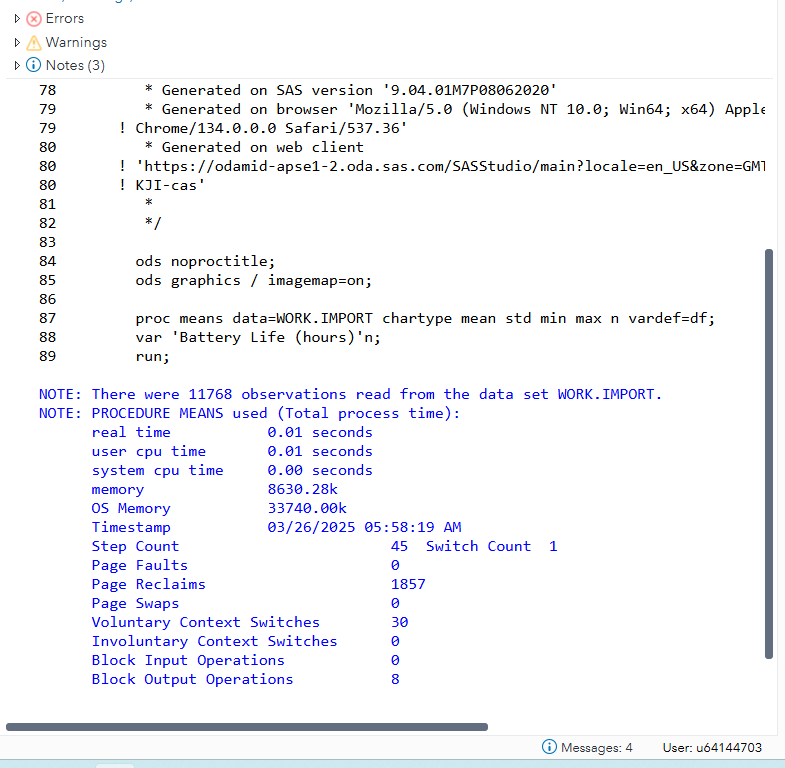
ods graphics / imagemap=on;

proc means data=WORK.IMPORT chartype mean std min max n vardef=df;

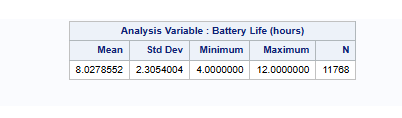
var 'Battery Life (hours)'n;

run;

**LOG:**

****

**OUTPUT:**

****

**PROC UNIVARIATE:**

**CODE:**

ods noproctitle;

ods graphics / imagemap=on;

/\* Exploring Data \*/

proc univariate data=WORK.IMPORT;

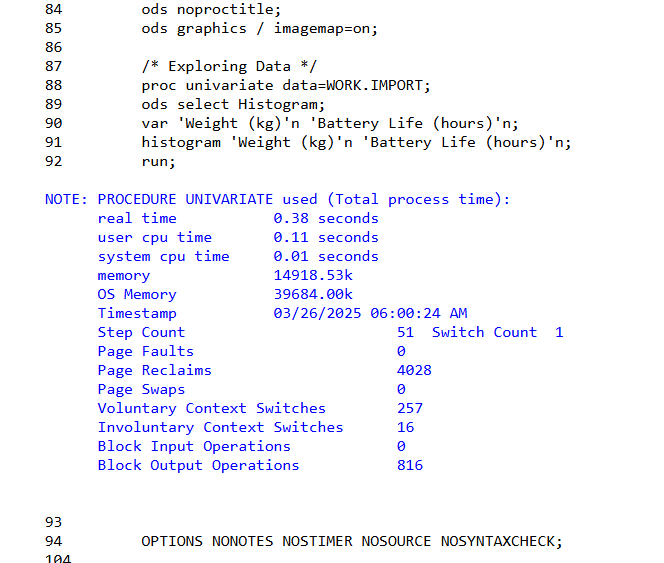
ods select Histogram;

var 'Weight (kg)'n 'Battery Life (hours)'n;

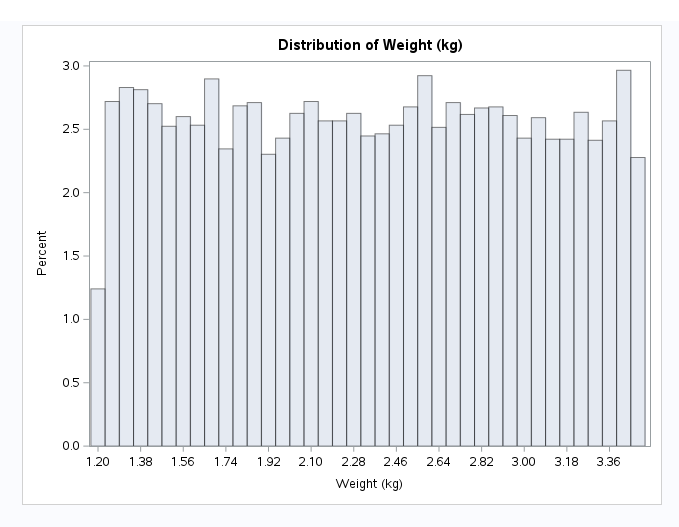
histogram 'Weight (kg)'n 'Battery Life (hours)'n;

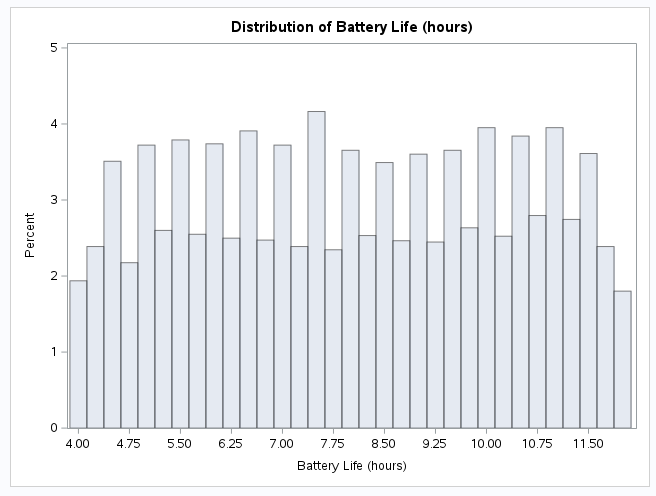
run;

**LOG:**

****

**OUTPUT:**

****

****