/*purpose: generate descriptive statistics, conduct hypothesis test and build model for comparison of numeric outcome variable between treatment groups*/
/*parameter:
1. input: input dataset name
2. output: output dataset name
3. variable: outcome variable name (numeric)
4. cohort: treatment variable name (numeric, value=0/1)
5. ref: reference level of treatment variable
6. questionlabel: label of outcome variable
7. tab_order: display order in table (numeric)
8. id: identify subject in the input dataset
9. surgeon: whether need to adjust for surgeon id (value=0/1, default is 0 - which means no)
10. surgeon_cluster: whether need to further cluster surgeon id(value=0/1, default is 0 - which means no)
11. surg_bar: if surgeon_cluster=1, degree of clustering surgeon id (numeric, higher is greater, default is 0)
12. diffs: whether need to estimate differences of predicted margin between treatment groups (value=0/1, means no/yes)
13. gee_linear: linear regression with GEE(value=0/1, default is 0 - which means no model building)
14. linear: linear regression(value=0/1, default is 0 - which means no model building)
15. pois: poisson regression with GEE (value=0/1, default is 0 - which means no model building)
16. ttest: t-test (value=0/1, default is 0 - which means no test)
17. anova: ANOVA (value=0/1, default is 0 - which means no test)
18. wilcoxon_rank_sum: wilcoxon rank sum test(value=0/1, default is 0 - which means no test)
19. kruskal_wallis: kruskal wallis test (value=0/1, default is 0 - which means no test)

output dataset variable:
1. answerlabel: row label of analysis block
2. answerorder: display order of row within the analysis block
3. tab_order: display order of analysis block

```
4. cohort_0: descriptive statistics of cohort=0
5. cohort_1: descriptive statistics of cohort=1
6. pvalue: p-value
7. diffs: difference of predicted margin between treatment groups
*/

%macro
stat_num(input,output,variable,cohort,ref,questionlabel,tab_order,id,surgeon=0,surgeon_cluster=0,surg_bar=0,diffs=0,gee_li
near=0,linear=0,pois=0,ttest=0,anova=0,wilcoxon_rank_sum=0,kruskal_wallis=0);
/*descriptive statistics*/
proc means data=&input stackodsoutput n mean std median min max q1 q3;
        class &cohort;
        var &variable;
        ods output summary=nout;
run;
data nout(drop=n Mean stddev min max q1 q3);
        length cohortlabel n_ mean_std min_max median_q1q3 $200;
        set nout;
        cohortlabel='cohort_'||strip(put(&cohort,best.));
        tab_order=&tab_order;
        N_=strip(put(N,best.));
        mean_std=strip(put(round(mean,0.01),10.2))||' ('||strip(put(round(stddev,0.01),10.2))||')';
        min_max=strip(put(round(min,0.1),10.1))||', '||strip(put(round(max,0.1),10.1));
        median_q1q3=strip(put(round(median,0.1),10.1))||' ('||strip(put(round(q1,0.1),10.1))||',
'||strip(put(round(q3,0.1),10.1))||')';
run;
proc sort data=nout; by &cohort cohortlabel tab_order; run;
```

```
proc transpose data=nout out=nout_;
	by &cohort cohortlabel tab_order;
	var n_ mean_std min_max median_q1q3;
run;
proc sort data=nout_; by tab_order _name_; run;
proc transpose data=nout_ out=nout_1;
	by tab_order _name_;
	var col1;
	id cohortlabel;
run;
data nout_2 (drop=_name_);
	length answerlabel $200.;
	set nout_1;
	if upcase(_name_)='N_' then do; answerlabel='N'; answer_order=1; end;
	if index(upcase(_name_),'MEAN') then do; answerlabel='Mean (SD)'; answer_order=2; end;
	if index(upcase(_name_),'MEDIAN') then do; answerlabel='Median (Q1, Q3)'; answer_order=3; end;
	if index(upcase(_name_),'MIN') then do; answerlabel='Min, Max'; answer_order=4; end;
run;

/*hypothesis test or model building part*/
%if &gee_linear=1 %then %do;
	proc genmod data=&input;
	class &cohort(ref="&ref") &id %if &surgeon=1 %then %do; sisicert3 %end;;
	model &variable = &cohort  %if &surgeon=1 %then %do; sisicert3 %end; / type3 dist=normal;
	repeated sub=&id /type=ind covb corrw;
	ods output type3=temp_pval(where=(upcase(source)="%upcase(&cohort)") rename=(ProbChiSq=pval))
	%if &diffs=1 %then %do; diffs=temp_diffs %end;;
```

```
        %if &diffs=1 %then %do; lsmeans &cohort / e diff cL; %end;
        run;
%end;
%if &linear=1 %then %do;
        proc genmod data=&input;
        class &cohort(ref="&ref") %if &surgeon=1 %then %do; sisicert3 %end;;
        model &variable = &cohort  %if &surgeon=1 %then %do; sisicert3 %end; / type3 dist=normal;
        ods output type3=temp_pval(where=(upcase(source)="%upcase(&cohort)") rename=(ProbChiSq=pval))
        %if &diffs=1 %then %do; diffs=temp_diffs %end;;
        %if &diffs=1 %then %do; lsmeans &cohort / e diff cL; %end;
        run;
%end;
%if &pois=1 %then %do ;

%if &surgeon=1 and &surgeon_cluster=1 %then %do;
        proc freq data=&input;
                tables sisicert3*&variable / out=surgeon;
        run;
        proc sql;
                create table surgeon_ as
                select distinct sisicert3,
                                                min(count) as fl1,
                                                count(distinct &variable) as fl2
                from surgeon(where=(sisicert3^='' and &variable is not missing))
                group by sisicert3;
        quit;
```

```
	proc sql;
		create table input as
		select a.*,
			case when a.sisicert2^='' and (fl2<=1 or (fl1<=&surg_bar and fl2=2)) then 'Other'
				else a.sisicert3 end as sisicert4
		from &input as a
		left join surgeon_ as b
		on a.sisicert3=b.sisicert3;
	quit;
	proc sql;
		select distinct sisicert3
		from input
		where sisicert3^=sisicert4;
	quit;
%end;
%if &surgeon=0 or &surgeon_cluster=0 %then %do;
	data input;
		length sisicert4 $200;
		set &input;
		sisicert4=sisicert3;
	run;
%end;
	proc genmod data=input ;
	class &cohort(ref="&ref") &id %if &surgeon=1 %then %do; sisicert4 %end;;
	model &variable = &cohort %if &surgeon=1 %then %do; sisicert4 %end;/ type3 dist = poisson;
	repeated sub=&id/type=ind covb corrw;
	ods output Type3  = temp_pval(where=(upcase(source)="%upcase(&cohort)") rename=(ProbChiSq=pval));
```

```
    run;

    %if &diffs=1 %then %do;
    %if &surgeon=1 %then %do;
    %margins(data=input,
class=&cohort &id sisicert4,
      response=&variable,
      model=&cohort sisicert4,
      dist=Poisson,
geesubject=&id,
      geecorr=IND,
margins=&cohort,
balanced=sisicert4,
diff=ALL,
options=cl);
    %end;
    %if &surgeon=0 %then %do;
    %margins(data=input,
class=&cohort &id,
      response=&variable,
      model=&cohort,
      dist=Poisson,
geesubject=&id,
      geecorr=IND,
margins=&cohort,
diff=ALL,
options=cl);
```

```
        %end;
        data temp_diffs;
                set _diffspm;
        run;
        %end;
%end;
%if &ttest=1 %then %do ;
        proc univariate data=&input normal plot;
        class &cohort;
        var &variable;
        run;

        proc ttest data=&input;
        class &cohort;
        var &variable;
        ods output Equality=Equality TTests=TTests;
        run;

        proc sql;
        create table temp_pval as
        select b.Probt as pval
        from Equality as a,TTests as b
        where (.<a.ProbF<0.05 and b.Method='Satterthwaite') or (a.ProbF>=0.05 and b.Method='Pooled');
        quit;
%end;
%if &anova=1 %then %do;
        proc anova data=&input;
```

```
	class &cohort;
	model &variable=&cohort;
	means &cohort / hovtest welch TUKEY;
	ods output HOVFTest=HOVFTest Welch=Welch ModelANOVA=ModelANOVA /*Means=means*/;
	run;

	/*Shapiro Wilk Test [Sample Size <= 50]
	Kolmogorov-Smirnov Test [Sample Size > 50];*/
	proc univariate data=/*residual*/&input normal plot;
	class &cohort;
	var /*residual*/&variable;
	run;

	proc sql;
	create table temp_pval as
	select case when .<a.ProbF<0.05 then c.ProbF
				when a.ProbF>=0.05 then b.ProbF end as pval
	from HOVFTest(where=(source=effect)) as a, ModelANOVA as b, Welch(where=(source=effect)) as c;
	quit;
%end;
/*use nonparametric test when normality assumption is not met*/
%if &wilcoxon_rank_sum=1 %then %do;
	proc npar1way data=&input wilcoxon;
	class &cohort;
	var &variable;
	ods output WilcoxonTest=temp_pval;
	/*use normal approximation when n1 and n2>=10, otherwise use exact wilcoxon*/
```

```
        *exact wilcoxon;
        run;

        data temp_pval;
        set temp_pval;
        rename Prob2=pval;
        run;
%end;
%if &kruskal_wallis=1 %then %do;
        proc npar1way data=&input wilcoxon DSCF;
        class &cohort;
        var &variable;
        ods output KruskalWallisTest=temp_pval;
        run;

        data temp_pval;
        set temp_pval;
        rename prob=pval;
        run;
%end;
/*combine outcome*/
options mprint;
%if &ttest=1 or &anova=1 or &wilcoxon_rank_sum=1 or &kruskal_wallis=1 or &gee_linear=1 or &linear=1 or
&pois=1 %then %do;
proc sql;
        create table pre_out as
        select m.*,
```

```
                %if &diffs=1 %then %do; strip(put(round(d.Estimate,0.01),10.2))||' ('||strip(put(round(d.Lower,0.01),10.2))||',
'||strip(put(round(d.Upper,0.01),10.2))||')' as diffs, %end;
                g.pval as pvalue format = pval.
        from nout_2 as m, temp_pval as g
        %if &diffs=1 %then %do; , temp_diffs as d %end;;
quit;
%end;
%if &ttest ne 1 and &anova ne 1 and &wilcoxon_rank_sum ne 1 and &kruskal_wallis ne 1 and &gee_linear ne 1 and &linear ne 1
and &pois ne 1 %then %do;
data pre_out;
        set nout_2;
run;
%end;
%if %bquote(&questionlabel) eq %then %do;
data &output; set pre_out; run;
%end;
%if %bquote(&questionlabel) ne %then %do;
data ql;
        format answerlabel $200.;
        tab_order=&tab_order;
        answerlabel="&questionlabel";
        answer_order=0;
run;
data &output;
        format answerlabel $200.;
        set ql pre_out(in=a);
        if a then do;
```

```
                answerlabel='  '||strip(answerlabel);
        end;
run;
%end;
proc datasets lib=work nolist ;
        delete input;
                quit ;
%mend;
```