/*purpose: generate descriptive statistics, conduct hypothesis test and build model for comparison of character outcome variable between treatment groups*/
/*parameter:
1. input: input dataset name
2. output: output dataset name
3. unit: value=eye/pat, indicate the record to be eye level or patient level
4. variable: outcome variable name(character)
5. min: for outcome variable, each level corresponds to a number, which indicates the display order of this level in table, show minimum number here
6. max: for outcome variable, each level corresponds to a number, which indicates the display order of this level in table, show maximum number here
7. cohort: treatment variable name (numeric, value=0/1)
8. ref: reference level of treatment variable
9. event: event category of outcome variable for the binary response model.
10. riskcol: for chi-square test and fisher test, if request risk difference, set the risk column=1/2
11. questionlabel: label of outcome variable
12. tab_order: display order of analysis block in table (numeric)
13. value_label: display format of outcome variable, link the displayed label in table with the outcome variable (eg: 0, is displayed as 'No', 1 is displayed as 'Yes')
14. order_label: for outcome variable, each level corresponds to a number (&min to &max), which indicates the display order of this level in table,
display format of order variable, link the displayed label in table with the order variable (&min to &max)
(eg: order of 0 is 1, which is displayed as 'No', order of 1 is 2, which is displayed as 'Yes', order variable value=1/2 is linked with 'No'/'Yes')
15. id: identify subject in the input dataset
16. self_denom: value=0/1, default is 0, which means derive denominator based on dataset for the calculation of percentage if 1(yes), need to manually provide denominator value for percentage calculation.

if unit=eye

denominator value for percentage calculation in cohort=0 column - store in &t0_eye

denominator value for percentage calculation in cohort=1 column - store in &t1_eye

if unit=pat

denominator value for percentage calculation in cohort=0 column - store in &t0_pat

denominator value for percentage calculation in cohort=1 column - store in &t1_pat

17. N: whether calculate the N, value=0/1, default is 0 - which means no
18. surgeon: whether need to adjust for surgeon id (value=0/1, default is 0 - which means no)
19. surgeon_cluster=0: whether need to further cluster surgeon id(value=0/1, default is 0 - which means no)
20. surg_bar: if surgeon_cluster=1, degree of clustering surgeon id (numeric, higher is greater, default is 0)
21. diffs: whether need to estimate differences of predicted margin between treatment groups (value=0/1, means no/yes)
22. gee_binomial: logistic regression with GEE (value=0/1, default is 0 - which means no model building)
23. gee_pois: poisson regression with GEE (value=0/1, default is 0 - which means no model building)
24. gee_multinomial: Cumulative logit model with GEE (value=0/1, default is 0 - which means no model building)
25. reverse: logistic regression model with GEE, reverse outcome variable and treatment variable in the model (value=0/1, default is 0 - which means no model is built)
26. chi: chi-square test (value=0/1, default is 0 - which means no test)
27. fisher: fisher exact test (value=0/1/2, default is 0 - which means no test , 1 for n*m table, 2 for 2*2 table)

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_char(input,output,unit,variable,min,max,cohort,ref,event,riskcol,questionlabel,tab_order,value_label,order_label,id,self_
denom=0,N=0,surgeon=0,surgeon_cluster=0,surg_bar=0,diffs=0,gee_binomial=0,gee_pois=0,gee_multinomial=0,reverse=0,c
hi=0,fisher=0);
/*determin the denominator for percentage calculation*/
%if &self_denom = 1 %then %do;
	%if &unit eq %str(eye) %then %do;
	proc sql;
		select count(*) into :t0_eye
		from &input
		where &cohort = 0 and &variable is not missing;
		select count(*) into :t1_eye
		from &input
		where &cohort = 1 and &variable is not missing;
	quit;
	%end;
	%if &unit eq %str(pat) %then %do;
	proc sql;
		select count(distinct &id ) into :t0_pat
		from &input
		where &cohort = 0 and &variable is not missing;
		select count(distinct &id ) into :t1_pat
		from &input
		where &cohort = 1 and &variable is not missing;
```

```
        quit;
        %end;
%end;
%if &unit eq %str(eye) %then %do;
        %let denominator0 = &t0_eye;
        %let denominator1 = &t1_eye;
%end;
%else %if &unit eq %str(pat) %then %do;
        %let denominator0 = &t0_pat;
        %let denominator1 = &t1_pat;
%end;
/*descriptive statistics calculation*/
%do i = &min %to &max;
        proc sql noprint;
                create table t0_&i. as
                select  count(*) as num_0,
                count(*)*100/ &denominator0 format = 6.2 as percent_0
                from &input
                where &cohort = 0 and put(&variable,&value_label..) = put(&i,&order_label..) ;

                create table t1_&i. as
                select  count(*) as num_1,
                count(*)*100/&denominator1 format = 6.2 as percent_1
                from &input
                where &cohort = 1 and put(&variable,&value_label..) = put(&i,&order_label..) ;

                create table dat&i. as
```

```
            select num_0,num_1,percent_0,percent_1
            from t0_&i.,t1_&i.;
        quit;

        data dat&i.;
        set dat&i.;
        format answerlabel $200.;
        cohort_0 = strip(put(num_0,best.))||' (' || strip(put(percent_0,10.2))||'%)';
        cohort_1 =strip(put(num_1,best.))||' (' || strip(put(percent_1,10.2))||'%)';
        tab_order=&tab_order;
        answerlabel=strip(put(&i,&order_label..));
        answer_order=&i;
        %if &N eq 1 %then %do;
            denominator_0=&denominator0;
            denominator_1=&denominator1;
        %end;
        drop num_0 num_1 percent_:;
        run;
%end;

data out_;
        set dat: ;
run;
proc sort data = out_;
        by answer_order;
run;
```

```
/*hypothesis test or model building part*/
%if &gee_binomial=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 (fl1<=&surg_bar or 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 descending;
    class &cohort(ref="&ref") &id %if &surgeon=1 %then %do; sisicert4 %end;;
    model &variable = &cohort %if &surgeon=1 %then %do; sisicert4 %end; / type3 dist=bin;
    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=&id &cohort sisicert4,
      response=&variable,
      roptions=event="&event",
      model= &cohort sisicert4,
      dist=binomial,
      geesubject=&id,
      geecorr=IND,
```

```
    margins=&cohort,
    balanced=sisicert4,
          diff=ALL,
    options=cl);
        %end;
        %if &surgeon=0 %then %do;
          %margins(data=input,
    class=&id &cohort,
          response=&variable,
          roptions=event="&event",
          model= &cohort,
          dist=binomial,
          geesubject=&id,
          geecorr=IND,
    margins=&cohort,
          diff=ALL,
    options=cl);
        %end;
        data temp_diffs;
                length diffs $200;
                set _diffspm;
                diffs=strip(put(round(Estimate,0.01),10.2))||' ('||strip(put(round(Lower,0.01),10.2))||',
'||strip(put(round(Upper,0.01),10.2))||')';
        run;
        %end;
%end;
```

```
%if &gee_pois=1 %then %do;
	proc genmod data=&input;
	class &cohort(ref="&ref") &id;
	model &variable = &cohort / type3 dist=poisson;
	repeated sub=&id /type=ind covb corrw;
	ods output Type3 = temp_pval(rename=(ProbChiSq=pval));
	run;
%end;
%if &gee_multinomial=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 rorder=internal;
        class &cohort(ref="&ref") &id %if &surgeon=1 %then %do; sisicert4 %end;;
        model &variable = &cohort %if &surgeon=1 %then %do; sisicert4 %end;/ type3 dist=MULTINOMIAL link=cumlogit;
        repeated sub=&id /type=ind covb corrw;
        ods output Type3 = temp_pval(where=(upcase(source)="%upcase(&cohort)") rename=(ProbChiSq=pval));
        run;
%end;
%if &reverse=1 %then %do ;
        proc genmod data=&input descending;
```

```
        class &id &cohort(ref="&ref") &variable;
        model &cohort = &variable / dist=bin type3;
        repeated sub=&id/type=ind covb corrw;
        ods output Type3  = temp_pval(rename=(ProbChiSq=pval));
        run;
%end ;
%if &chi=1 %then %do ;
        proc freq data = &input;
        tables &cohort*&variable / chisq expected norow nocol nopercent
        %if &diffs=1 %then %do; riskdiff(CL=WALD column=&riskcol) %end;;
        ods output ChiSq = temp_pval (where=(statistic="Chi-Square") rename=(prob=pval))
        %if &diffs=1 %then %do; PdiffCLs=PdiffCLs %end;;
        run;
        %if &diffs=1 %then %do;
        data temp_diffs;
                length diffs $200;
                set PdiffCLs;
                diffs=strip(put(round(-RiskDifference,0.01),10.2))||' ('||strip(put(round(-UpperCL,0.01),10.2))||',
'||strip(put(round(-LowerCL,0.01),10.2))||')';
        run;
        %end;
%end ;
/*if = 20% of expected cell counts are less than 5, then use the chi-square test;
if > 20% of expected cell counts are less than 5, then use Fisher's exact test*/
%if &fisher=1 %then %do ;   /*n*m*/
        proc freq data = &input;
        tables &cohort*&variable / fisher expected norow nocol nopercent ;
```

```
        ods output FishersExact = temp_pval (where=(Label1="Pr <= P") rename=(nValue1=pval));
        run;
%end ;
%if &fisher=2 %then %do ;   /*2*2*/
        proc freq data = &input;
        tables &cohort*&variable / fisher expected norow nocol nopercent
        %if &diffs=1 %then %do; riskdiff(CL=WALD column=&riskcol) %end;;
        /*%if &diffs=1 %then %do; exact riskdiff; %end; better to use the CL=EXACT, but it takes too long to run*/
        ods output FishersExact = temp_pval (where=(Label1="Two-sided Pr <= P") rename=(nValue1=pval))
        %if &diffs=1 %then %do; PdiffCLs=PdiffCLs %end;;
        run;
        %if &diffs=1 %then %do;
        data temp_diffs;
                length diffs $200;
                set PdiffCLs;
                diffs=strip(put(round(-RiskDifference,0.01),10.2))||' ('||strip(put(round(-UpperCL,0.01),10.2))||',
'||strip(put(round(-LowerCL,0.01),10.2))||')';
        run;
        %end;
%end ;
/*combine outcome*/
%if &chi=1 or &fisher=1 or &fisher=2 or &gee_binomial=1 or &gee_multinomial=1 or &reverse=1 or &gee_pois=1 %then %do;
proc sql;
        create table pre_out as
        select m.*,
                %if &diffs=1 %then %do; d.diffs, %end;
                g.pval as pvalue format = pval.
```

```
        from out_ as m, temp_pval as g
        %if &diffs=1 %then %do; , temp_diffs as d %end;;
quit;
%end;
%if &chi ne 1 and &fisher ne 1 and &fisher ne 2 and &gee_binomial ne 1 and &gee_multinomial ne 1 and &reverse ne 1 and
&gee_pois ne 1 %then %do;
data pre_out;
        set out_;
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=&min -1;
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 dat: t0_: t1_: input;
                quit ;
%mend;
```