Module 1

This code processes columns, creates some additional columns using first. and last., and summarizes data.

```
libname work list;

libname dir '/folders/myshortcuts/analysis_files';

data use_events (drop = timestamp_micros);
        set dir.events
                (keep =  instance_id
                                _id
                                userid
                                kns_lid
                                timestamp_micros
                                name

                                kns__screen_class
                                kns__previous_class

                                campaign
                                medium
                                source

                                kns__conversion
                                kns__event_origin
                 rename = (name = event_name));

        /* Convert EPOCH date time to readable format
                EPOCH is considered from 01Jan70
                SAS considers datetime from 01Jan60
                Adding 01Jan60 to 01Jan70 time difference and using datetime format */

                datetime_stamp = int(timestamp_micros/1000000) + "01jan1970 0:0:0"dt;
                Format datetime_stamp datetime.;

                date_stamp = datepart(datetime_stamp);
                Format date_stamp Date9.;

run;

proc sort data = use_events; by instance_id date_stamp datetime_stamp; run;

data use_events2 (drop = temp prev_date_stamp);
        set use_events (drop = _id);
        by instance_id date_stamp;

        retain temp;
        prev_date_stamp = lag1(date_stamp);

        if first.instance_id then do;
                temp = 1;
                session = 1;
        end;
        else if date_stamp EQ prev_date_stamp then do;
                session = temp;
        end;
        else if date_stamp NE prev_date_stamp then do;
                session = temp + 1;
                temp = session;
        end;

run;

proc sql;
        create table use_events3 as
        select a.*
        from use_events2 as a

        join (
                select instance_id, session, count(*) as dummy
                from use_events2
                where event_name EQ 'session_start'
                group by instance_id, session
        ) as b
```

```
                on a.instance_id = b.instance_id and a.session = b.session
                ;
        quit;



/* Merge user dim and screen view, with userid and kns_lid from SIGN_IN event */

        data _user_kns_lid (drop = event_name);
                set use_events3 (keep = instance_id session datetime_stamp userid kns_lid event_name);
                where event_name = "SIGN_IN";
                proc sort; by instance_id session descending datetime_stamp;
                proc sort nodupkey; by instance_id session;
        run;

        proc sql;
                create table use_events4 as
                select a.*, b.userid, b.kns_lid
                from use_events3 (drop = userid kns_lid) as a

                left join _user_kns_lid as b
                on a.instance_id = b.instance_id and a.session = b.session
                ;
        quit;



/* First and Last touch (at user level, due to defined path) */

        data _touch1;
                set use_events4 (keep = instance_id session event_name datetime_stamp campaign medium source);
                where event_name = "kns__campaign";
                if campaign EQ "" and medium EQ "" and source EQ "" then delete;
        run;

        proc freq data = _touch1; tables source source * medium * campaign /list missing; run;

        * First touch touch;
                proc sort data = _touch1; by instance_id session datetime_stamp; run;

                data _touch_first;
                        set _touch1;
                        by instance_id session datetime_stamp;

                        *if first.instance_id then output _touch_first;
                        *else output _touch_rest;

                        if first.instance_id then output;

                run;

        * Last touch touch;
                proc sort data = _touch1; by instance_id descending session descending datetime_stamp; run;

                data _touch_last;
                        set _touch1;
                        by instance_id descending session descending datetime_stamp;

                        if first.instance_id then output;
                run;

        * Combine datasets;
                proc sort data = _touch_first nodupkey; by instance_id; run;
                proc sort data = _touch_last nodupkey; by instance_id; run;

                data use_touch;
                        merge _touch_first (in = a keep = instance_id campaign medium source
                                                                                rename = (campaign = FirstTouch_campaign
medium = FirstTouch_medium source = FirstTouch_source))

                                        _touch_last (in = b keep = instance_id campaign medium source
                                                                                rename = (campaign = LastTouch_campaign medium
= LastTouch_medium source = LastTouch_source));

                        by instance_id;
                        if a or b;
                run;
```

```
/* First time */

        data use_firsttime_opens (drop = event_name);
                set use_events4 (keep = instance_id event_name);
                where event_name = "first_open";
                first_open = 1;
                proc sort nodupkey; by instance_id;
        run;



/* Count of sessions */

        proc sql;
                create table use_sessions_count as
                select instance_id, count(distinct session) as count_sessions
                from use_events4
                group by instance_id;
        quit;

        proc freq data = use_sessions_count; tables count_sessions; run;


        data use_events_auto (drop = kns__event_origin)
                 use_events_web (drop = kns__event_origin);

                set use_events4;
                drop campaign medium source kns__conversion;

                where event_name NOT IN ('kns__campaign' 'session_start' 'SIGN_IN' 'first_open');

                if event_name EQ "web_remove" then
                        kns__screen_class = "web_remove";

                if kns__event_origin = "auto" then output use_events_auto;
                else if kns__event_origin = "web" then output use_events_web;

        run;

        proc sort data = use_events_auto; by instance_id session datetime_stamp; run;
        proc sort data = use_events_web; by instance_id session datetime_stamp; run;


        proc sql;

                create table use_events_rolled_hit_level as
                select instance_id,
                                sum(case when event_name = 'SIGN_IN' then 1 else . end) as event_SIGN_IN,
                                sum(case when event_name = 'SELECT_LANG' then 1 else . end) as event_SELECT_LANG,
                                sum(case when event_name = 'GETTING_STARTED' then 1 else . end) as
event_GETTING_STARTED,
                                sum(case when event_name = 'HOW_IT_WORKS' then 1 else . end) as event_HOW_IT_WORKS,
                                sum(case when event_name = 'web_remove' then 1 else . end) as event_web_REMOVE

                from use_events4
                where kns__event_origin = "web" or event_name ="web_remove"

                group by instance_id
                order by instance_id
                ;


                create table use_events_rolled_user_level as
                select instance_id,
                                count(distinct case when event_name = 'SIGN_IN' then 1 else . end) as event_SIGN_IN,
                                count(distinct case when event_name = 'SELECT_LANG' then 1 else . end) as
event_SELECT_LANG,
                                count(distinct case when event_name = 'GETTING_STARTED' then 1 else . end) as
event_GETTING_STARTED,
                                count(distinct case when event_name = 'HOW_IT_WORKS' then 1 else . end) as
event_HOW_IT_WORKS,
                                count(distinct case when event_name = 'web_remove' then 1 else . end) as
event_web_REMOVE

                from use_events4
                where kns__event_origin = "web" or event_name ="web_remove"
```

```
                group by instance_id
                order by instance_id
                ;

        quit;

        data use_events_rolled;
                set use_events_rolled_hit_level (in = a)
                        use_events_rolled_user_level (in = b);

                if a then level = "Hit";
                else if b then level = "User";

                proc sort; by instance_id level; run;
        run;


/* User dimensions*/

        data use_userdims;
                set dir.user_dim;

                keep instance_id
                         bundle_sequence_id

                         user_acquired_medium
                         user_acquired_source
                         user_acquired_campaign


                         ;

                proc sort; by instance_id descending bundle_sequence_id;
                proc sort nodupkey; by instance_id;
        run;

        proc sort data = use_touch nodupkey; by instance_id; run;
        proc sort data = use_sessions_count nodupkey; by instance_id; run;
        proc sort data = use_firsttime_opens nodupkey; by instance_id; run;
        proc sort data = use_userdims nodupkey; by instance_id; run;

        data use_userdims2;
                merge   use_userdims (in = a)
                                use_touch (in = b)
                                use_sessions_count (in = c)
                                use_firsttime_opens (in = d);
                by instance_id;
                if a;

                proc sort; by instance_id; run;
        run;


        data final_events_rolled;
                merge use_events_rolled (in = a)
                          use_userdims2 (in = b);
                by instance_id;
                if a;
        run;

        proc summary data = final_events_rolled nway missing;
                class   level
                                count_sessions

                                user_acquired_medium
                                user_acquired_source
                                user_acquired_campaign

                                first_open

                                FirstTouch:
                                LastTouch:
                                ;

                var     event: ;
                output out = smry_final_events_rolled (drop = _type_ rename = (_freq_ = count_users)) sum=;
        run;
```

```
        proc export data = smry_final_events_rolled
                outfile = '/folders/myshortcuts/analysis_files/output/smry_final_events_rolled.csv'
                dbms = csv replace;
        run;

/* Transpose */

        proc transpose data = work.use_events4 out = work.use_events4_tposed prefix = screen_seq_;
                where name = "screen_view";
                by instance_id;
                var kns__screen_class;
        run;
```