SAS Code for mobility analysis

```
/******************************************************************************************
 Copyright OCS Life Sciences
*******************************************************************************************


**********************************************************************************;

/* Read and manipulate Google data */

**********************************************************************************;


data INSAS.GOOGLE    ;
  %let _EFIERR_ = 0; /* set the ERROR detection macro variable */
  infile "&filepath.\Global_Mobility_Report.csv" delimiter = ',' MISSOVER DSD lrecl=32767
firstobs=2 ;
   informat country_region_code $2. ;
   informat country $50. ;
   informat sub_region_1 $50. ;
   informat sub_region_2 $50. ;
   informat date ddmmyy10. ;
   informat retail_and_recreation best32. ;
   informat grocery_and_pharmacy best32. ;
   informat parks best32. ;
   informat transit_stations best32. ;
   informat workplaces best32. ;
   informat residential best32. ;
   format country_region_code $2. ;
   format country $50. ;
   format sub_region_1 $50. ;
   format sub_region_2 $50. ;
   format date ddmmyy10. ;
   format retail_and_recreation best12. ;
   format grocery_and_pharmacy best12. ;
   format parks best12. ;
   format transit_stations best12. ;
   format workplaces best12. ;
   format residential best12. ;
  input
            country_region_code  $
            country  $
            sub_region_1  $
            sub_region_2  $
            date
            retail_and_recreation
            grocery_and_pharmacy
            parks
            transit_stations
            workplaces
            residential
;
if _ERROR_ then call symputx('_EFIERR_',1);  /* set ERROR detection macro variable */
run;

/* Format date and rename variables and change unit of measure to % from 100% */
data work.google1;
  set insas.google;
    format date ddmmyy10.;
      if sub_region_1 = "" and sub_region_2 = "";
    array transform{*} retail_and_recreation grocery_and_pharmacy parks transit_stations
workplaces residential;
      do i = 1 to 6;
        if transform(i) ne . then transform(i) = transform(i) + 100;
      end;
run;

data adam.google(drop = country_region_code i);
  set work.google1;
    if country = "Czechia" then country = "Czech Republic";
    if country = "Côte d'Ivoire" then country = "Cote d'Ivoire";
    if country = "Moldova" then country = "Moldova Republic Of";
    if country = "Myanmar (Burma)" then country = "Myanmar";
    if country = "North Macedonia" then country = "North Macedonia Republic Of";
    if country = "United States" then country = "United States of America";
    /* Calculate mean_mobility */
    if retail_and_recreation ne . or transit_stations ne . or workplaces ne . or
grocery_and_pharmacy ne . then mean_mobility = mean(retail_and_recreation, transit_stations,
workplaces, grocery_and_pharmacy);

    weekday = weekday(date);
    weekno = week(date);
run;
```

/* Read and manipulate Apple data */

```
*************************************************************************;
```

```
data insas.apple    ;
   %let _EFIERR_ = 0; /* set the ERROR detection macro variable */
   infile "&filepath.\applemobilitytrends-2020-06-14.csv" delimiter = ',' MISSOVER DSD
lrecl=32767 firstobs=2 ;
      informat geo_type $50. ;
      informat country $50. ;
      informat transportation_type $20. ;
      informat sub_region $50.;
      informat country_spec $50.;
      informat _13_01_2020 best32. ;
      informat _14_01_2020 best32. ;
      informat _15_01_2020 best32. ;
      informat _16_01_2020 best32. ;
      informat _17_01_2020 best32. ;
      informat _18_01_2020 best32. ;
      informat _19_01_2020 best32. ;
      informat _20_01_2020 best32. ;
      informat _21_01_2020 best32. ;
      informat _22_01_2020 best32. ;
      informat _23_01_2020 best32. ;
      informat _24_01_2020 best32. ;
      informat _25_01_2020 best32. ;
      informat _26_01_2020 best32. ;
      informat _27_01_2020 best32. ;
      informat _28_01_2020 best32. ;
      informat _29_01_2020 best32. ;
      informat _30_01_2020 best32. ;
      informat _31_01_2020 best32. ;
      informat _01_02_2020 best32. ;
      informat _02_02_2020 best32. ;
      informat _03_02_2020 best32. ;
      informat _04_02_2020 best32. ;
      informat _05_02_2020 best32. ;
      informat _06_02_2020 best32. ;
         informat _07_02_2020 best32. ;
         informat _08_02_2020 best32. ;
         informat _09_02_2020 best32. ;
         informat _10_02_2020 best32. ;
         informat _11_02_2020 best32. ;
         informat _12_02_2020 best32. ;
         informat _13_02_2020 best32. ;
         informat _14_02_2020 best32. ;
         informat _15_02_2020 best32. ;
         informat _16_02_2020 best32. ;
         informat _17_02_2020 best32. ;
         informat _18_02_2020 best32. ;
         informat _19_02_2020 best32. ;
         informat _20_02_2020 best32. ;
         informat _21_02_2020 best32. ;
         informat _22_02_2020 best32. ;
         informat _23_02_2020 best32. ;
         informat _24_02_2020 best32. ;
         informat _25_02_2020 best32. ;
         informat _26_02_2020 best32. ;
         informat _27_02_2020 best32. ;
         informat _28_02_2020 best32. ;
         informat _29_02_2020 best32. ;
         informat _01_03_2020 best32. ;
         informat _02_03_2020 best32. ;
```

```
 informat _03_03_2020 best32. ;
 informat _04_03_2020 best32. ;
 informat _05_03_2020 best32. ;
 informat _06_03_2020 best32. ;
 informat _07_03_2020 best32. ;
 informat _08_03_2020 best32. ;
 informat _09_03_2020 best32. ;
 informat _10_03_2020 best32. ;
 informat _11_03_2020 best32. ;
 informat _12_03_2020 best32. ;
 informat _13_03_2020 best32. ;
 informat _14_03_2020 best32. ;
 informat _15_03_2020 best32. ;
 informat _16_03_2020 best32. ;
 informat _17_03_2020 best32. ;
 informat _18_03_2020 best32. ;
 informat _19_03_2020 best32. ;
 informat _20_03_2020 best32. ;
 informat _21_03_2020 best32. ;
 informat _22_03_2020 best32. ;
 informat _23_03_2020 best32. ;
 informat _24_03_2020 best32. ;
 informat _25_03_2020 best32. ;
 informat _26_03_2020 best32. ;
 informat _27_03_2020 best32. ;
 informat _28_03_2020 best32. ;
 informat _29_03_2020 best32. ;
 informat _30_03_2020 best32. ;
informat _31_03_2020 best32. ;
informat _01_04_2020 best32. ;
informat _02_04_2020 best32. ;
informat _03_04_2020 best32. ;
informat _04_04_2020 best32. ;
informat _05_04_2020 best32. ;
informat _06_04_2020 best32. ;
informat _07_04_2020 best32. ;
informat _08_04_2020 best32. ;
informat _09_04_2020 best32. ;
informat _10_04_2020 best32. ;
informat _11_04_2020 best32. ;
informat _12_04_2020 best32. ;
informat _13_04_2020 best32. ;
informat _14_04_2020 best32. ;
informat _15_04_2020 best32. ;
informat _16_04_2020 best32. ;
informat _17_04_2020 best32. ;
informat _18_04_2020 best32. ;
informat _19_04_2020 best32. ;
informat _20_04_2020 best32. ;
informat _21_04_2020 best32. ;
informat _22_04_2020 best32. ;
informat _23_04_2020 best32. ;
informat _24_04_2020 best32. ;
informat _25_04_2020 best32. ;
informat _26_04_2020 best32. ;
informat _27_04_2020 best32. ;
informat _28_04_2020 best32. ;
informat _29_04_2020 best32. ;
informat _30_04_2020 best32. ;
informat _01_05_2020 best32. ;
informat _02_05_2020 best32. ;
informat _03_05_2020 best32. ;
informat _04_05_2020 best32. ;
informat _05_05_2020 best32. ;
informat _06_05_2020 best32. ;
informat _07_05_2020 best32. ;
informat _08_05_2020 best32. ;
informat _09_05_2020 best32. ;
informat _10_05_2020 best32. ;
informat _11_05_2020 best32. ;
informat _12_05_2020 best32. ;
informat _13_05_2020 best32. ;
informat _14_05_2020 best32. ;
informat _15_05_2020 best32. ;
informat _16_05_2020 best32. ;
informat _17_05_2020 best32. ;
informat _18_05_2020 best32. ;
informat _19_05_2020 best32. ;
informat _20_05_2020 best32. ;
informat _21_05_2020 best32. ;
informat _22_05_2020 best32. ;
informat _23_05_2020 best32. ;
informat _24_05_2020 best32. ;
informat _25_05_2020 best32. ;
informat _26_05_2020 best32. ;
informat _27_05_2020 best32. ;
informat _28_05_2020 best32. ;
informat _29_05_2020 best32. ;
informat _30_05_2020 best32. ;
informat _31_05_2020 best32. ;
informat _01_06_2020 best32. ;
informat _02_06_2020 best32. ;
informat _03_06_2020 best32. ;
informat _04_06_2020 best32. ;
informat _05_06_2020 best32. ;
informat _06_06_2020 best32. ;
informat _07_06_2020 best32. ;
informat _08_06_2020 best32. ;
informat _09_06_2020 best32. ;
informat _10_06_2020 best32. ;
informat _11_06_2020 best32. ;
informat _12_06_2020 best32. ;
informat _13_06_2020 best32. ;
informat _14_06_2020 best32. ;

format geo_type $50. ;
format country $50. ;
format transportation_type $20. ;
format sub_region $50.;
format country_spec $50.;
format _13_01_2020 best12. ;
format _14_01_2020 best12. ;
format _15_01_2020 best12. ;
format _16_01_2020 best12. ;
format _17_01_2020 best12. ;
format _18_01_2020 best12. ;
format _19_01_2020 best12. ;
format _20_01_2020 best12. ;
format _21_01_2020 best12. ;
format _22_01_2020 best12. ;
format _23_01_2020 best12. ;
format _24_01_2020 best12. ;
format _25_01_2020 best12. ;
format _26_01_2020 best12. ;
format _27_01_2020 best12. ;
format _28_01_2020 best12. ;
format _29_01_2020 best12. ;
format _30_01_2020 best12. ;
format _31_01_2020 best12. ;
format _01_02_2020 best12. ;
format _02_02_2020 best12. ;
format _03_02_2020 best12. ;
format _04_02_2020 best12. ;
format _05_02_2020 best12. ;
format _06_02_2020 best12. ;
format _07_02_2020 best12. ;
format _08_02_2020 best12. ;
format _09_02_2020 best12. ;
format _10_02_2020 best12. ;
format _11_02_2020 best12. ;
format _12_02_2020 best12. ;
format _13_02_2020 best12. ;
format _14_02_2020 best12. ;
format _15_02_2020 best12. ;
format _16_02_2020 best12. ;
format _17_02_2020 best12. ;
format _18_02_2020 best12. ;
format _19_02_2020 best12. ;
format _20_02_2020 best12. ;
format _21_02_2020 best12. ;
 format _22_02_2020 best12. ;
 format _23_02_2020 best12. ;
 format _24_02_2020 best12. ;
 format _25_02_2020 best12. ;
 format _26_02_2020 best12. ;
 format _27_02_2020 best12. ;
 format _28_02_2020 best12. ;
 format _29_02_2020 best12. ;
 format _01_03_2020 best12. ;
 format _02_03_2020 best12. ;
 format _03_03_2020 best12. ;
 format _04_03_2020 best12. ;
 format _05_03_2020 best12. ;
 format _06_03_2020 best12. ;
 format _07_03_2020 best12. ;
 format _08_03_2020 best12. ;
 format _09_03_2020 best12. ;
 format _10_03_2020 best12. ;
 format _11_03_2020 best12. ;
 format _12_03_2020 best12. ;
 format _13_03_2020 best12. ;
 format _14_03_2020 best12. ;
 format _15_03_2020 best12. ;
 format _16_03_2020 best12. ;
 format _17_03_2020 best12. ;
 format _18_03_2020 best12. ;
 format _19_03_2020 best12. ;
 format _20_03_2020 best12. ;
 format _21_03_2020 best12. ;
 format _22_03_2020 best12. ;
 format _23_03_2020 best12. ;
 format _24_03_2020 best12. ;
 format _25_03_2020 best12. ;
 format _26_03_2020 best12. ;
 format _27_03_2020 best12. ;
 format _28_03_2020 best12. ;
 format _29_03_2020 best12. ;
 format _30_03_2020 best12. ;
 format _31_03_2020 best12. ;
 format _01_04_2020 best12. ;
 format _02_04_2020 best12. ;
 format _03_04_2020 best12. ;
 format _04_04_2020 best12. ;
 format _05_04_2020 best12. ;
 format _06_04_2020 best12. ;
 format _07_04_2020 best12. ;
 format _08_04_2020 best12. ;
 format _09_04_2020 best12. ;
 format _10_04_2020 best12. ;
 format _11_04_2020 best12. ;
 format _12_04_2020 best12. ;
 format _13_04_2020 best12. ;
 format _14_04_2020 best12. ;
 format _15_04_2020 best12. ;
 format _16_04_2020 best12. ;
 format _17_04_2020 best12. ;
 format _18_04_2020 best12. ;
 format _19_04_2020 best12. ;
 format _20_04_2020 best12. ;
 format _21_04_2020 best12. ;
 format _22_04_2020 best12. ;
 format _23_04_2020 best12. ;
 format _24_04_2020 best12. ;
 format _25_04_2020 best12. ;
 format _26_04_2020 best12. ;
 format _27_04_2020 best12. ;
 format _28_04_2020 best12. ;
 format _29_04_2020 best12. ;
 format _30_04_2020 best12. ;
 format _01_05_2020 best12. ;
 format _02_05_2020 best12. ;
 format _03_05_2020 best12. ;
 format _04_05_2020 best12. ;
 format _05_05_2020 best12. ;
 format _06_05_2020 best12. ;
 format _07_05_2020 best12. ;
 format _08_05_2020 best12. ;
 format _09_05_2020 best12. ;
 format _10_05_2020 best12. ;
 format _11_05_2020 best12. ;
 format _12_05_2020 best12. ;
```

```
   format _13_05_2020 best12. ;
   format _14_05_2020 best12. ;
   format _15_05_2020 best12. ;
   format _16_05_2020 best12. ;
  format _17_05_2020 best32. ;
  format _18_05_2020 best32. ;
  format _19_05_2020 best32. ;
  format _20_05_2020 best32. ;
  format _21_05_2020 best32. ;
  format _22_05_2020 best32. ;
  format _23_05_2020 best32. ;
  format _24_05_2020 best32. ;
  format _25_05_2020 best32. ;
  format _26_05_2020 best32. ;
  format _27_05_2020 best32. ;
  format _28_05_2020 best32. ;
  format _29_05_2020 best32. ;
  format _30_05_2020 best32. ;
  format _31_05_2020 best32. ;
  format _01_06_2020 best32. ;
  format _02_06_2020 best32. ;
  format _03_06_2020 best32. ;
  format _04_06_2020 best32. ;
  format _05_06_2020 best32. ;
  format _06_06_2020 best32. ;
  format _07_06_2020 best32. ;
  format _08_06_2020 best32. ;
  format _09_06_2020 best32. ;
  format _10_06_2020 best32. ;
  format _11_06_2020 best32. ;
  format _12_06_2020 best32. ;
  format _13_06_2020 best32. ;
  format _14_06_2020 best32. ;
input
            geo_type  $
            country  $
            transportation_type  $
            sub_region $
            country_spec $
            _13_01_2020
            _14_01_2020
            _15_01_2020
            _16_01_2020
            _17_01_2020
            _18_01_2020
            _19_01_2020
            _20_01_2020
            _21_01_2020
            _22_01_2020
            _23_01_2020
            _24_01_2020
            _25_01_2020
            _26_01_2020
            _27_01_2020
            _28_01_2020
            _29_01_2020
            _30_01_2020
            _31_01_2020
            _01_02_2020
            _02_02_2020
            _03_02_2020
            _04_02_2020
            _05_02_2020
            _06_02_2020
           _07_02_2020
           _08_02_2020
           _09_02_2020
           _10_02_2020
           _11_02_2020
           _12_02_2020
           _13_02_2020
           _14_02_2020
           _15_02_2020
           _16_02_2020
           _17_02_2020
           _18_02_2020
           _19_02_2020
```

```
_20_02_2020
_21_02_2020
_22_02_2020
_23_02_2020
_24_02_2020
_25_02_2020
_26_02_2020
_27_02_2020
_28_02_2020
_29_02_2020
_01_03_2020
_02_03_2020
_03_03_2020
_04_03_2020
_05_03_2020
_06_03_2020
 _07_03_2020
 _08_03_2020
 _09_03_2020
 _10_03_2020
 _11_03_2020
 _12_03_2020
 _13_03_2020
 _14_03_2020
 _15_03_2020
 _16_03_2020
 _17_03_2020
 _18_03_2020
 _19_03_2020
 _20_03_2020
 _21_03_2020
 _22_03_2020
 _23_03_2020
 _24_03_2020
 _25_03_2020
 _26_03_2020
 _27_03_2020
 _28_03_2020
 _29_03_2020
 _30_03_2020
 _31_03_2020
 _01_04_2020
 _02_04_2020
 _03_04_2020
 _04_04_2020
 _05_04_2020
_06_04_2020
_07_04_2020
_08_04_2020
_09_04_2020
_10_04_2020
_11_04_2020
_12_04_2020
_13_04_2020
_14_04_2020
_15_04_2020
_16_04_2020
_17_04_2020
_18_04_2020
_19_04_2020
_20_04_2020
_21_04_2020
_22_04_2020
_23_04_2020
_24_04_2020
_25_04_2020
_26_04_2020
_27_04_2020
_28_04_2020
_29_04_2020
_30_04_2020
_01_05_2020
_02_05_2020
_03_05_2020
_04_05_2020
_05_05_2020
_06_05_2020
```

```
                 _07_05_2020
                 _08_05_2020
                 _09_05_2020
                 _10_05_2020
                 _11_05_2020
                 _12_05_2020
                 _13_05_2020
                 _14_05_2020
                 _15_05_2020
                 _16_05_2020
                 _17_05_2020
                 _18_05_2020
                 _19_05_2020
                 _20_05_2020
                 _21_05_2020
                 _22_05_2020
                 _23_05_2020
                 _24_05_2020
                 _25_05_2020
                 _26_05_2020
                 _27_05_2020
                 _28_05_2020
                 _29_05_2020
                 _30_05_2020
                 _31_05_2020
                 _01_06_2020
                 _02_06_2020
                 _03_06_2020
                 _04_06_2020
                 _05_06_2020
                 _06_06_2020
                 _07_06_2020
                 _08_06_2020
                 _09_06_2020
                 _10_06_2020
                 _11_06_2020
                 _12_06_2020
                 _13_06_2020
                 _14_06_2020

     ;
  if _ERROR_ then call symputx('_EFIERR_',1);  /* set ERROR detection macro variable */
run;

proc sort data=insas.apple out=work.apple1 (where = (geo_type = "country/region"));
  by geo_type country sub_region country_spec;
run;

proc transpose data=work.apple1 out=work.apple2;
  by geo_type country sub_region country_spec;
  id transportation_type;
run;

/* Converting dates into date field */

data work.apple3;
  set work.apple2;
    day = input(substr(_name_,2,2),8.0);
    month = input(substr(_name_,5,2),8.0);
    year = input(substr(_name_,8,4),8.0);
    date = mdy(month,day,year);
    format date ddmmyy10.;
run;


/* Flag baseline value */

data work.apple4(keep = country date baseflag driving transit walking);
   set work.apple3;
     if date = 21927 then baseflag = 1;
     if country = "Russia" then country = "Russian Federation";
     if country = "UK" then country = "United Kingdom";
     if country = "United States" then country = "United States of America";
run;


/* Calculate the mean mobility of walking, driving and transit */
```

```
data adam.apple;
  set work.apple4;
    if driving ne . or transit ne . or walking ne . then mean_mobility =
mean(driving,transit,walking);
    weekday = weekday(date);
    weekno = week(date);
run;
************************************************************************************;
```

/* Read and manipulate NPI data */

```
************************************************************************************;


/* Import Government intervention data (NPI) */

data insas.NPI    ;
    %let _EFIERR_ = 0; /* set the ERROR detection macro variable */
    infile "&filepath.\acaps_covid19_government_measures_dataset.csv" delimiter = ',' MISSOVER
DSD lrecl=32767 firstobs=2 ;
       informat ID best32. ;
       informat COUNTRY $50. ;
       informat ISO $3. ;
       informat REGION $50. ;
       informat LOGTYPE $100. ;
       informat CATEGORY $100. ;
       informat MEASURE $100. ;
       informat TARPOPGR $3. ;
       informat DATE_IMP ddmmyy10. ;
       informat STYPE $10. ;
       format ID best32. ;
       format COUNTRY $50. ;
       format ISO $3. ;
       format REGION $50. ;
       format LOGTYPE $100. ;
       format CATEGORY $100. ;
       format MEASURE $100. ;
       format TARPOPGR $3. ;
       format DATE_IMP ddmmyy10. ;
       format STYPE $10. ;

    input
                ID
                COUNTRY  $
                ISO  $
                REGION  $
                LOGTYPE  $
                CATEGORY  $
                MEASURE  $
                TARPOPGR  $
                DATE_IMP
                STYPE  $
       ;
  if _ERROR_ then call symputx('_EFIERR_',1);  /* set ERROR detection macro variable */
run;

/* Modify some countries/regions to merge with Google and Apple mobility data */
data work.npi1;
  set insas.npi;
    if tarpopgr = "No";  /* Only keep Nationally implemented interventions */
    if country = "Czech republic" then country = "Czech Republic";
    if country = "C��d'Ivoire" then country = "Cote d'Ivoire";
    if country = "kenya" then country = "Kenya";
    if country = "Viet Nam" then country = "Vietnam";

    if measure = "limit public gatherings" then measure = "Limit public gatherings";
    if measure = "testing policy" then measure = "Testing policy";
    if measure = "curfews" then measure = "Curfews";
    if measure = "strengthening the public health system" then measure = "Strengthening the
public health system";
    if measure = "Border checks�" then measure = "Border checks";
    if measure = "Border closure�" then measure = "Border closure";
    if measure = "Schools closure�" then measure = "Schools closure";
    if measure = "awareness campaigns" then measure = "Awareness campaigns";
    if measure = "full lockdown" then measure = "Full lockdown";
    if measure = "requirement to wear protective gear in public" then measure = "Requirement
to wear protective gear in public";

run;

/* Determine the start of each intervention per country */

proc sort data=work.npi1 out=work.npi2 (where = (logtype = "Introduction / extension of
measures" and date_imp ne .));
  by country measure date_imp;
run;

data work.npi3;
  set work.npi2;
    by country measure date_imp;
      if first.measure then inter_start_fl = 1;
run;

data adam.npi;
  set work.npi3;
    length parm $20.;
   /* Create shorter descriptions */
   if measure = "Economic measures" then parm = "ECO_MEAS";
   if measure = "Emergency administrative structures activated or established" then parm =
"ADMIN";
   if measure = "Limit product imports/exports" then parm = "IMP_EXP";
   if measure = "Military deployment" then parm = "MIL_DEP";
   if measure = "State of emergency declared" then parm = "EMERGENCY";
   if measure = "Full lockdown" then parm = "FLOCK";
   if measure = "Partial lockdown" then parm = "PLOCK";
   if measure = "Border checks" then parm = "BOR_CHECK";
   if measure = "Border closure" then parm = "BOR_CLOSE";
   if measure = "Complete border closure" then parm = "CBOR_CLOSE";
   if measure = "Checkpoints within the country" then parm = "CHECK";
   if measure = "Curfews" then parm = "CURFEWS";
   if measure = "Domestic travel restrictions" then parm = "DOMESTIC";
   if measure = "International flights suspension" then parm = "INTERNAT";
   if measure = "Surveillance and monitoring" then parm = "MONITOR";
   if measure = "Visa restrictions" then parm = "VISA";
   if measure = "Amendments to funeral and burial regulations" then parm = "FUNERAL";
   if measure = "Awareness campaigns" then parm = "AWARE";
   if measure = "General recommendations" then parm = "GENERAL";
   if measure = "Health screenings in airports and border crossings" then parm = "HEALTH";
   if measure = "Isolation and quarantine policies" then parm = "POLICIES";
   if measure = "Mass population testing" then parm = "MASSTEST";
   if measure = "Obligatory medical tests not related to COVID-19" then parm = "MEDTEST";
   if measure = "Other public health measures enforced" then parm = "PUBHEALTH";
   if measure = "Psychological assistance and medical social work" then parm = "PSYCH";
   if measure = "Requirement to wear protective gear in public" then parm = "PROTGEAR";
   if measure = "Strengthening the public health system" then parm = "HEALTHSYS";
   if measure = "Testing policy" then parm = "TEST";
   if measure = "Changes in prison-related policies" then parm = "PRISON";
   if measure = "Closure of businesses and public services" then parm = "BUS_CLOSE";
   if measure = "Limit public gatherings" then parm = "PUB_GATHER";
   if measure = "Schools closure" then parm = "SCHOOL";
   if measure = "Humanitarian exemptions" then parm = "HUMAN";
   if measure = "Lockdown of refugee/idp camps or other minorities" then parm = "RLOCK";
   if measure = "Additional health/documents requirements upon arrival" then parm = "DOCS";
run;


***************************************************************************************;

/* Smooth weekday effect */

***************************************************************************************;


%macro smooth(indata=,param=,init_cutoff=,outdata=);

data work.smooth;
```

```
  set &indata.;
    if &param. < &init_cutoff.;
run;

proc sort data=work.smooth;
  by country date;
run;

data work.smooth1(keep = country date rename = (date=first_day_below_70));
  set work.smooth;
    by country date;
      if first.country;
run;

proc sort data=&indata.;
  by country;
run;

data work.smooth2;
  merge &indata. work.smooth1;
    by country;
run;

data work.smooth3;
  set work.smooth2;
    if date>=first_day_below_70 and weekday = 1;
run;

proc sort data=work.smooth3;
  by country date;
run;

data work.smooth4(keep = country date rename = (date=first_sunday_below_70));
  set work.smooth3;
    by country date;
      if first.country;
run;

data work.smooth5;
  merge work.smooth2 work.smooth4;
    by country;
run;

data work.smooth6;
  set work.smooth5;
    if date >= first_sunday_below_70 then smooth_flag = 1;
run;


/* Smooth the timeseries to account for seasonality */
/* Calculate average for each week */
proc sort data=work.smooth6 (where = (smooth_flag=1));
  by country weekno;
run;

proc means data=work.smooth6 noprint;
  by country weekno;
    var &param.;
    output out=work.mean_week
    mean = mean_week;
run;

data work.smooth7;
  merge work.smooth6 work.mean_week;
    by country weekno;
run;

/* Calculate SD for each weekday from the average of the week */
data work.smooth8;
  set work.smooth7;
    if &param. ne . and mean_week ne . then weekday_SD = (mean_week - &param.);
run;

/* Calculate the mean SD for each weekday */

proc sort data=work.smooth8;
  by country weekday;
```

```
run;

proc means data=work.smooth8 noprint;
  by country weekday;
    var weekday_SD;
      output out=work.weekday_SD_mean
      mean = weekday_SD_mean;
run;

data work.smooth9;
  merge work.smooth8 work.weekday_SD_mean;
    by country weekday;
run;

/* ADD the mean SD for each weekday from the original mobility */

data work.smooth10(keep = country date &param. first_day_below_70 first_sunday_below_70
smooth_flag mean_week weekday_SD weekday_SD_mean new_&param.);
  set work.smooth9;
    if &param. ne . and weekday_SD_mean ne . then new_&param. = &param. + weekday_SD_mean;
run;

proc sort data=work.smooth10;
  by country date &param.;
run;

proc sort data=&indata.;
  by country date &param.;
run;

data work.final;
  merge &indata. work.smooth10;
    by country date &param.;
run;

data &outdata.;
  set work.final;
    if smooth_flag ne 1 and &param. ne . then new_&param. = &param.;
run;

%mend;
```

```
******************************************************************************;
```

/* Merge Apple and Google data with NPI dataset */

```
******************************************************************************;

%macro merge_npi(indata=,param=,outdata=);
/* Merge data with NPI data */

data work.npi_select;
  set adam.npi;
    if inter_start_fl = 1;
run;


proc sort data=work.npi_select;
  by country;
run;

proc transpose data=work.npi_select out=work.npit (drop = _name_) ;
  by country;
  id parm;
  var date_imp;
run;

data work.temp(keep = country date &param. new_&param. mobility_change stable_start_date
adjusted_stable_start_date stable_&param. lost_&param.);
  set &indata.;
```

```
run;

proc sort data=work.npit;
  by country;
run;

proc sort data=work.temp;
  by country;
run;

data &outdata.;
  merge work.npit (in=x) work.temp (in=y);
    by country;
      if x and y;
run;

%mend;


***********************************************************************************;

/* Calculate average % mobility remaining */

***********************************************************************************;

%macro average_stable_mobility(indata=, param=, init_cutoff=, hcutoff=,
no_days_stable=,outdata=);


/* Check if mean mobility is below 70% and flag records */

data work.temp;
  set &indata.;
     if new_&param. < &init_cutoff. then flag = 1;
run;

proc sort data=work.temp out=work.amobile;
  by country date;
run;

data work.amobile1;
  set work.amobile;
     by country date;
     lag_mobility = lag(new_&param.);
     lag_date = lag(date);
     if first.country then do;
       lag_mobility = new_&param.;
       lag_date = date;
     end;
run;

/* Calculate the % change from previous previous day */

data work.amobile2;
  set work.amobile1;
    if lag_mobility > 0 and new_&param. ne . and lag_mobility ne . then mobility_change =
(new_&param. - lag_mobility);
    if date ne . and lag_date ne . then diff_date = date - lag_date;
run;

/* Count number of consecutive days where the mobility changed cutoff  */

data work.amobile3;
  set work.amobile2;
    by country date;
      if first.country then do;
        count = 0;
        reset = 0;
     end;
        if flag ne 1 or diff_date ne 1 or mobility_change >= &hcutoff. then do;
           count = 0;
        end;
        if flag = 1 and diff_date = 1 and mobility_change < &hcutoff. then count + 1;
        if count = 1 then reset + 1;
run;
```

```
data work.amobile4;
  set work.amobile3;
    if count = 0 then reset = .;
run;

/* Calculate the maximum number of days stable for each reset */

proc sort data=work.amobile4;
  by country reset;
run;

proc means data=work.amobile4 (where = (reset ne .)) noprint;
  by country reset;
  var count;
  output out=ano_days (keep = country reset no_of_days_stable)
    max = no_of_days_stable;
run;

data work.amobile5;
  merge work.amobile4 work.ano_days;
    by country reset;
run;

/* Get the first stabilizing date */

data work.astable;
  set work.amobile5;
    if no_of_days_stable >= &no_days_stable.;
run;

proc sort data=work.astable;
  by country date;
run;

data work.astable1(keep = country date rename = (date=stable_start_date));
  set work.astable;
    by country date;
      if first.country;
run;

data work.amobile6;
  merge work.amobile5 work.astable1;
    by country;
run;


/* Calculate mean mobility % over stabilized period */

data work.stable_period;
  set work.amobile6;
    if stable_start_date ne . and date ne . and stable_start_date <= date < (stable_start_date
+ 14) and flag = 1;
run;

proc sort data=work.stable_period;
  by country stable_start_date;
run;

proc means data =work.stable_period noprint;
  by country stable_start_date;
  var &param.;
  output out=work.amean_mobility (keep = country stable_&param.)
    mean = stable_&param.;
run;

/* Merge stabilizing date with google data */

data work.temp2;
  merge work.amobile6 work.amean_mobility;
    by country;
run;


/* Adjust stabilizing date to the first day the mobility falls below the stable mean mobility
*/
```

```
proc sort data=work.temp2 (where = (stable_&param. ne . and new_&param. ne . and
new_&param.<=stable_&param. and flag = 1 and no_of_days_stable >= 4)) out=work.new_stable;
  by country date;
run;

data work.new_stable1 (keep = country date rename = (date=adjusted_stable_start_date));
  set work.new_stable;
    by country date;
      if first.country;
run;

data work.temp3;
  merge work.temp2 work.new_stable1;
    by country;
run;

/* Calculate the average mobility lost */

data &outdata.;
  set work.temp3;
    if stable_&param. ne . then lost_&param. = 100 - stable_&param.;
run;

proc sort data=&outdata.;
  by country date;
run;

%mend;
```

***;

/* Calculate Slope / decline */

***;

```
/* Plot individual countries with all calculated parameters shown on graph */

%macro plotdata(indata=,param=,outfile=,max=);

proc sort data=&indata.;
  by country date;
run;

ods pdf file   = "../TLF output/&outfile..pdf" style = journal bookmarklist=hide nogtitle
nogfootnote;
ods graphics / width = 1300 border=off;
options nobyline nodate nonumber;
Title "#Byval(country)";

proc sgplot data=&indata.;
     by country;
    scatter x=date y=&param. ;

    series x=date y=new_&param. /name="Weekday" legendlabel="Corrected weekday effect
mobility";
    series x=date y=&param. / lineattrs=(color =green) name="Avgmob" legendlabel="Mean
Mobility %";
    series x=date y=stable_&param. / lineattrs=(color = blue) name="Mean" legendlabel="Average
% remaining mobility";
    *refline stable_start_date / name="Stable" axis=x lineattrs=(color = red)
LEGENDLABEL="Stable Start Date";
    refline adjusted_stable_start_date / name="AdjStable" axis=x lineattrs=(color = red
pattern = 2)  LEGENDLABEL="Adjusted stable mobility date";
    refline decline_start / axis=x lineattrs=(color = green pattern = 2) name="Decline"
legendlabel = "Start of decline";

    yaxis label="Mean Mobility %" min=0 max=&max.;
    xaxis label="Date";
    keylegend "Avgmob" / location=outside position=bottom noborder;
    keylegend "Weekday" / location=outside position=bottom noborder;
    keylegend "AdjStable" / location=outside position=bottom noborder;
    keylegend "Mean" /  location=outside position=bottom noborder;
    keylegend "Decline" /  location=outside position=bottom noborder;
run;

ods pdf close;

%mend;


%plotdata(indata=adam.npi_apple_decline,param=mean_mobility,outfile=Apple_Mean_mobility_per_co
untry_decline,max=280);


/* Calculating the gradient from the start of decline to the start of stable graph */

data work.start_decline(keep = country date mean_mobility rename = (date=x1
mean_mobility=y1));
  set adam.npi_apple_decline;
    if date = decline_start;
run;

data work.start_stable(keep = country date mean_mobility rename = (date=x2 mean_mobility=y2));
  set adam.npi_apple_decline;
    if date = adjusted_stable_start_date;
run;

data work.slope;
  merge work.start_decline work.start_stable;
     by country;
run;

data work.slope1(keep = country slope days_decline_stable);
  set work.slope;
    slope = (y2-y1)/(x2-x1);
    days_decline_stable = (x2-x1)+1;
run;


data adam.npi_apple_decline;
  merge adam.npi_apple_decline work.slope1;
    by country;
run;


/* Calculate descriptive statistics on slope and days between the start of decline and
stability */

ods pdf file   = "../TLF output/Apple_slope_stats.pdf" style = journal bookmarklist=hide
nogtitle nogfootnote;
title1 "Descriptive statistics for the slope between the start of decline and the start of
stable mobility";

proc sort data=adam.npi_apple_decline out=npi_apple nodupkey;
  by country slope days_decline_stable;
run;

proc means data=npi_apple n mean std median min max lclm uclm;
  var slope;
  output out=slope_stats
    n = n
    mean = mean
    std =std
    min = min
    median = median
    max = max
    lclm = lclm
    uclm = uclm;
run;

proc univariate data=npi_apple plots normal;
  var slope;
run;
```

```
ods pdf close;


/*  Regression analysis on the slope */


data work.apple_analysis;
  set adam.apple_analysis;
run;

proc sort data=adam.npi_apple_decline out=work.slope (keep = country slope
days_decline_stable) nodupkey;
  by country slope days_decline_stable;
run;

data adam.apple_analysis;
  merge work.apple_analysis work.slope;
    by country;
run;


/* New analysis - setting any intervention that happened more than 3 weeks before the
stabilizing point to 0 */

data work.new_analysis;
  set adam.npi_apple_decline;
    if flock ne . and adjusted_stable_start_date ne . and flock <= adjusted_stable_start_date
then lock = flock;
       else if plock ne . and adjusted_stable_start_date ne . and plock <=
adjusted_stable_start_date and lock = . then lock = plock;
     format lock;

    array create_var{*} aware bor_check bor_close prison domestic admin general health
internat policies imp_exp pub_gather pubhealth
                        plock school emergency healthsys monitor check bus_close curfews
eco_meas mil_dep flock protgear docs
                        rlock masstest test visa funeral psych human cbor_close medtest lock;
      do i = 1 to 36;
        if create_var(i) ne . and adjusted_stable_start_date - 21 <= create_var(i) <=
adjusted_stable_start_date then create_var(i) = 1;
          else create_var(i) =0;
      end;

      format aware bor_check bor_close prison domestic admin general health internat policies
imp_exp pub_gather pubhealth
                        plock school emergency healthsys monitor check bus_close curfews
eco_meas mil_dep flock protgear docs
                        rlock masstest test visa funeral psych human cbor_close medtest lock;
run;

proc sort data=work.new_analysis out=work.new_analysis1 nodupkey;
  by country adjusted_stable_start_date;
run;


ods pdf file   = "../TLF output/Apple_analysis_output_slope_occurrence_disregarding all NPIs
before 3 weeks.pdf" style = journal bookmarklist=hide nogtitle nogfootnote;

title1 "Slope: Regression model on occurrence of intervention (disregarding all NPIs before 3
weeks) - lowest AIC (including intercept)";

/* Exclude South Africa because it is an extreme outlier */
ods trace on;
proc glmselect data=work.new_analysis1 (where = (country not in ("South Africa") and slope ne
.));
  class aware bor_close admin general health internat policies pub_gather lock school
healthsys bus_close eco_meas visa emergency/  order=internal DESCENDING;
  model slope=aware bor_close admin general health internat policies pub_gather lock school
healthsys bus_close eco_meas visa emergency / select=aic showpvalues;
  output out=out p=p r=r;
run;
ods trace off;

title2 "Diagnostic plots to check for homogeneity and normality of the residuals of the best
fit model";
```

```
proc glm data=work.new_analysis1 (where = (country not in ("South Africa") and slope ne .))
PLOTS(UNPACK)=DIAGNOSTICS;
   class pub_gather lock school emergency;
   model slope=pub_gather lock school emergency;
run;

title1 "Slope: Regression model on occurrence of intervention - best fit according to highest
adjusted r-squared (including intercept)";
ods trace on;
proc glmselect data=work.new_analysis1 (where = (country not in ("South Africa") and slope ne
.));
  class aware bor_close admin general health internat policies pub_gather lock school
healthsys bus_close eco_meas visa emergency/  order=internal DESCENDING;
  model slope=aware bor_close admin general health internat policies pub_gather lock school
healthsys bus_close eco_meas visa emergency / select=adjrsq showpvalues;
  output out=out p=p r=r;
run;
ods trace off;


ods pdf close;

/*******************END OF APPLE DATA*************************************/;


*----------------------------------------------------------------------------------
                             GOOGLE DATA
----------------------------------------------------------------------------------;

%macro start_decline (indata=,var=,outdata=);
proc sort data=&indata. out=work.temp1;
  by country date;
run;

data work.temp2;
  set work.temp1;
    by country date;
      if mdy(2,15,2020) <= date <= adjusted_stable_start_date then before_fl = 1;
      lagbefore_fl = lag(before_fl);
      lag_mobility = lag(new_&var.);
      lag_date = lag(date);
      if first.country then do;
        lag_mobility = new_&var.;
        lag_date = date;
      end;
run;

data work.temp3;
  set work.temp2;
    if date ne . and lag_date ne . then diff_date = date - lag_date;
run;

data work.count_increase;
  set work.temp3;
    by country date;
      if first.country then do;
        count_increase = 0;
      end;
      if before_fl = 1 and -2 < mobility_change < 7 then count_increase + 1;
      if mobility_change <= -2 or mobility_change >=7 or before_fl = . then count_increase =
0;
run;


data work.decline_start ;
  set work.count_increase;
    by country date;
      if first.country then do;
        count = 0;
        reset = 0;
      end;
        if before_fl = 1 and mobility_change < 7 then do;
           count + 1;
           if count_increase >= 3 then count = 0;
        end;
      if mobility_change >= 7 or before_fl = . then count = 0;
      if count = 1 then reset + 1;
```

```
run;

data work.decline_start1;
  set work.decline_start;
    if count = 0 then reset = .;
run;

proc sort data=work.decline_start1;
  by country reset date;
run;

data work.reset_change (keep = country reset sum_mobility_change mobility_start mobility_end);
  set work.decline_start1;
    by country reset date;
      retain mobility_start;
      if first.reset then do;
        sum_mobility_change = mobility_change;
        mobility_start = lag_mobility;
      end;
        else sum_mobility_change + mobility_change;
      if last.reset then mobility_end = new_&var.;
      if last.reset;
run;

data work.reset_change1;
  set work.reset_change;
    calc_mobility_change = mobility_end - mobility_start;
run;

data work.decline_start2;
  merge work.decline_start1 work.reset_change1;
    by country reset;
run;

/* Calculate the maximum for each reset */
proc sort data=work.decline_start2;
  by country reset;
run;

data work.decline_start3;
  set work.decline_start2;
    by country reset;
      if last.reset and count >=1 then more_than_x_decline = 1;
run;

data work.no_days_decline(keep = country reset more_than_x_decline rename =
(more_than_x_decline=more_than_x_decline_fl));
  set work.decline_start3;
    if more_than_x_decline = 1;
run;

data work.decline_start4;
  merge work.decline_start3 work.no_days_decline;
    by country reset;
run;

/* Calculate the start of decline date */

data work.decline;
  set work.decline_start4;
    if more_than_x_decline_fl = 1 and calc_mobility_change <= -(0.60 * lost_&var.);
run;

/* Get the first date for each reset */
proc sort data=work.decline;
  by country reset date;
run;

data work.first_reset(keep = country date first_reset_fl reset);
  set work.decline;
    by country reset date;
      if first.reset;
      first_reset_fl = 1;
run;

proc sort data=work.first_reset;
  by country reset;
```

```
run;

data work.first_reset1;
  set work.first_reset;
    by country reset;
      if last.country;
      decline_start = date - 1;
      format decline_start date9.;
run;

/* Get the last date per reset to check if the mean_mobility after the drop is below 80% */
proc sort data=work.decline;
  by country reset date;
run;

data work.last_reset(keep = country last_reset_fl reset);
  set work.decline;
    by country reset date;
      if last.reset and &var. < 80 ;
      last_reset_fl = 1;
run;

data work.first_reset2;
  merge work.first_reset1 (in=x) work.last_reset;
    by country reset;
      if x;
run;

data &outdata.;
  merge work.temp3 work.first_reset2 (keep = country decline_start) ;
    by country;
run;

proc sort data=&outdata.;
  by country date;
run;

%mend;

%start_decline
(indata=adam.npi_google_rar,var=retail_and_recreation,outdata=adam.npi_google_rar_decline);
%plotdata(indata=adam.npi_google_rar_decline,param=retail_and_recreation,outfile=Google_RAR_Me
an_mobility_per_country_decline,max=160);

%start_decline
(indata=adam.npi_google_ts,var=transit_stations,outdata=adam.npi_google_ts_decline);
%plotdata(indata=adam.npi_google_ts_decline,param=transit_stations,outfile=Google_TS_Mean_mobi
lity_per_country_decline,max=140);

/* Calculating the slope from the start of decline to the start of stable graph */

%macro slope(indata=,var=,outdata=);


data work.start_decline(keep = country date &var. rename = (date=x1 &var.=y1));
  set &indata.;
    if date = decline_start;
run;

data work.start_stable(keep = country date &var. rename = (date=x2 &var.=y2));
  set &indata.;
    if date = adjusted_stable_start_date;
run;

data work.slope;
  merge work.start_decline work.start_stable;
     by country;
run;

data work.slope1(keep = country slope days_decline_stable);
  set work.slope;
    slope = (y2-y1)/(x2-x1);
    days_decline_stable = (x2-x1)+1;
run;


data &outdata.;
```

```
  merge &indata. work.slope1;
    by country;
run;

%mend;


%slope(indata=adam.npi_google_rar_decline,var=new_retail_and_recreation,outdata=adam.npi_googl
e_rar_decline_slope);

/* Calculate descriptive statistics on slope and days between the start of decline and
stability */
ods pdf file   = "../TLF output/Google RAR/Google_RAR_slope_stats.pdf" style = journal
bookmarklist=hide nogtitle nogfootnote;
title1 "Descriptive statistics for the slope between the start of decline and the start of
stable mobility";

proc sort data=adam.npi_google_rar_decline_slope out=npi_google_rar nodupkey;
  by country slope days_decline_stable;
run;

proc means data=npi_google_rar (where = (slope ne . )) n mean std median min max lclm uclm;
  var slope;
  output out=slope_stats
    n = n
    mean = mean
    std =std
    min = min
    median = median
    max = max
    lclm = lclm
    uclm = uclm;
run;

proc univariate data=npi_google_rar (where = (slope ne . )) plots normal;
  var slope;
run;

ods pdf close;


%slope(indata=adam.npi_google_ts_decline,var=new_transit_stations,outdata=adam.npi_google_ts_d
ecline_slope);

/* Calculate descriptive statistics on slope and days between the start of decline and
stability */
ods pdf file   = "../TLF output/Google TS/Google_TS_slope_stats.pdf" style = journal
bookmarklist=hide nogtitle nogfootnote;
title1 "Descriptive statistics for the slope between the start of decline and the start of
stable mobility";

proc sort data=adam.npi_google_ts_decline_slope out=npi_google_ts nodupkey;
  by country slope days_decline_stable;
run;

proc means data=npi_google_ts (where = (slope ne . )) n mean std median min max lclm uclm;
  var slope;
  output out=slope_stats
    n = n
    mean = mean
    std =std
    min = min
    median = median
    max = max
    lclm = lclm
    uclm = uclm;
run;

proc univariate data=npi_google_ts (where = (slope ne . )) plots normal;
  var slope;
run;

ods pdf close;

/*******************END OF GOOGLE DATA*************************************/;
```

```
**********************************************************************************;

/* NPI Cluster and associations analysis */

**********************************************************************************;


*-----------------------------------------------------------------------------------
                             APPLE DATA
------------------------------------------------------------------------------------;


proc sort data=work.new_analysis out=work.new_analysis1 nodupkey;
  by country adjusted_stable_start_date;
run;

proc means data=work.new_analysis1 n mean std median mode min max;
  var no_interventions_3weeks;
  output out=stat_no_interventions_3weeks
    n = n
    mean = mean
    std = std
    median = median
    mode = mode
    min = min
    max = max;
run;

/* Calculating Chi-square results */

%let v1 = aware;
%let v2 = bor_close;
%let v3 = admin;
%let v4 = general;
%let v5 = health;
%let v6 = internat;
%let v7 = policies;
%let v8 = pub_gather;
%let v9 = lock;
%let v10 = school;
%let v11 = emergency;
%let v12 = healthsys;
%let v13 = bus_close;
%let v14 = eco_meas;
%let v15 = visa;

data work.chisqr_output;
  length table $256. statistic $28. ;
  table = "";
  statistic = "";
  df = .;
  value = .;
  prob = .;
run;

options spool;
%macro chisquare(indata=);

%DO i=1 %TO 14;
  %DO j=(&i+1) %TO 15;
   ods output chisq = work.stat_output;
   PROC FREQ DATA = &indata. ;
      %STR(tables  &&v&i * &&v&j / chisq OUT= out_&i._&j.; );
   RUN;

    PROC DATASETS library=work force;
      APPEND BASE=work.chisqr_output  data=work.stat_output;
    QUIT;
    RUN;
   ods output close;
  %END;
%END;

%mend;


%chisquare(indata=work.new_analysis1);

data adam.Apple_int_chisqr_excNPI3weeks;
  set work.chisqr_output;
     if statistic = "Chi-Square";
     if prob < 0.05 then sig_flag = "Y";
run;

data assoc;
  set adam.Apple_int_chisqr_excNPI3weeks;
     length first second $50.;
     star = find(table,"*",1);
     first = substr(table,7,star-7);
     second = substr(table,star+1,30);
run;

proc sort data=assoc nodupkey;
  by first second prob;
run;

proc transpose data=assoc out=assoct;
  by first;
  id second;
  var prob;
run;

ods trace on;
proc corr data=work.new_analysis1;
  ods output PearsonCorr=corr_out;
  var aware bor_check bor_close prison domestic admin general health internat policies imp_exp
pub_gather pubhealth
                        school emergency healthsys monitor check bus_close curfews eco_meas
mil_dep protgear docs
                        rlock masstest test visa funeral psych human cbor_close medtest lock;
run;
ods trace off;

/*      PERFORM CLUSTER ANALYSIS ON THE ORDER VARIABLE */
/* Determine unique countries and label them 1 to 59 */

proc sort data=work.new_analysis1 out=work.countries (keep = country) nodupkey;
  by country;
run;

data work.countries1;
  set work.countries;
    index = 1;
run;

proc sort data=work.countries1;
  by index;
run;

data work.countries2;
  set work.countries1;
    by index;
      if first.index then countryn = 1;
        else countryn + 1;
run;

data work.new_analysis2;
  merge work.new_analysis1 work.countries2 (drop = index);
    by country;
run;

/* Transpose data */

proc transpose data=work.new_analysis2 out=work.new_analysis3 (rename = (_name_ = parm col1 =
ordervar));
  by country countryn;
  var AWARE BOR_CLOSE ADMIN GENERAL HEALTH INTERNAT POLICIES PUB_GATHER LOCK SCHOOL HEALTHSYS
BUS_CLOSE ECO_MEAS VISA EMERGENCY;
run;

proc sort data=work.new_analysis3;
  by parm;
run;

proc transpose data=work.new_analysis3 out=work.analysist prefix=country;
  by parm;
  id countryn;
  var ordervar;
run;


ods pdf file   = "../TLF output/Apple_cluster_analysis_output_excluding NPIs less than 3
weeks.pdf" style = journal bookmarklist=hide nogtitle nogfootnote;

title1 "Cluster analyis on all NPIs occurring within 3 weeks from sabilizing";

proc distance data=work.analysist method=djaccard absent=0 out=distjacc;
  var anominal(country1 country2 country3 country4 country5 country6 country7 country8
country9 country10
                country11 country12 country13 country14 country15 country16 country17
country18 country19 country20
                country21 country22 country23 country24 country25 country26 country27
country28 country29 country30
                country31 country32 country33 country34 country35 country36 country37
country38 country39 country40
                country41 country42 country43 country44 country45 country46 country47
country48 country49 country50
                country51 country52 country53 country54 country55 country56 country57
country58 country59);
  id parm;
run;


proc print data=distjacc;
  id parm;
run;

proc cluster data=distjacc method=centroid pseudo outtree=tree;
   id parm;
run;


ods pdf close;






*-----------------------------------------------------------------------------------
                             GOOGLE RAR DATA
-----------------------------------------------------------------------------------;


data work.new_rar_analysis;
  set adam.npi_google_rar;
    if flock ne . and adjusted_stable_start_date ne . and flock <= adjusted_stable_start_date
then lock = flock;
       else if plock ne . and adjusted_stable_start_date ne . and plock <=
adjusted_stable_start_date and lock = . then lock = plock;
     format lock;

    array create_var{*} aware bor_check bor_close prison domestic admin general health
internat policies imp_exp pub_gather pubhealth
                        plock school emergency healthsys monitor check bus_close curfews
eco_meas mil_dep flock protgear docs
                        rlock masstest test visa funeral psych human cbor_close medtest lock;
      do i = 1 to 36;
        if create_var(i) ne . and adjusted_stable_start_date - 21 <= create_var(i) <=
adjusted_stable_start_date then create_var(i) = 1;
          else create_var(i) =0;
      end;
      format aware bor_check bor_close prison domestic admin general health internat policies
imp_exp pub_gather pubhealth
                        plock school emergency healthsys monitor check bus_close curfews
eco_meas mil_dep flock protgear docs
                        rlock masstest test visa funeral psych human cbor_close medtest lock;

       no_interventions_3weeks = sum(aware, bor_check, bor_close, prison, domestic, admin,
general, health, internat, policies, imp_exp, pub_gather,
                            pubhealth, school, emergency, healthsys, monitor, check,
bus_close, curfews, eco_meas, mil_dep, lock,
                            protgear, docs, rlock, masstest, test, visa, funeral, psych,
human, cbor_close, medtest);


run;

proc sort data=work.new_rar_analysis out=work.new_rar_analysis1 (where = (country not in
("Jamaica", "Mongolia", "Togo"))) nodupkey;
  by country adjusted_stable_start_date;
run;


proc means data=work.new_rar_analysis1 n mean std median mode min max;
  var no_interventions_3weeks;
  output out=stat_no_interventions_3weeks
    n = n
    mean = mean
    std = std
    median = median
    mode = mode
    min = min
    max = max;
run;


/* Calculating Chi-square results */

%let v1 = aware;
%let v2 = bor_close;
%let v3 = domestic;
%let v4 = admin;
%let v5 = general;
%let v6 = health;
%let v7 = internat;
%let v8 = policies;
%let v9 = pub_gather;
%let v10 = lock;
%let v11 = school;
%let v12 = emergency;
%let v13 = healthsys;
%let v14 = bus_close;
%let v15 = curfews;
%let v16 = eco_meas;
%let v17 = visa;

data work.chisqr_output;
  length table $256. statistic $28. ;
  table = "";
  statistic = "";
  df = .;
  value = .;
  prob = .;
run;

options spool;
%macro chisquare(indata=);

%DO i=1 %TO 16;
  %DO j=(&i+1) %TO 17;
   ods output chisq = work.stat_output;
   PROC FREQ DATA = &indata. ;
      %STR(tables  &&v&i * &&v&j / chisq OUT= out_&i._&j.; );
   RUN;

    PROC DATASETS library=work force;
      APPEND BASE=work.chisqr_output  data=work.stat_output;
    QUIT;
    RUN;
   ods output close;

  %END;
%END;

%mend;


%chisquare(indata=work.new_rar_analysis1);

data adam.RAR_int_chisqr_excNPI3weeks;
  set work.chisqr_output;
     if statistic = "Chi-Square";
     if prob < 0.05 then sig_flag = "Y";
run;




/*      PERFORM CLUSTER ANALYSIS ON THE ORDER VARIABLE */
/* Determine unique countries and label them 1 to 121 */

proc sort data=work.new_rar_analysis1 out=work.countries (keep = country) nodupkey;
  by country;
run;

data work.countries1;
  set work.countries;
    index = 1;
run;

proc sort data=work.countries1;
  by index;
run;

data work.countries2;
  set work.countries1;
    by index;
      if first.index then countryn = 1;
        else countryn + 1;
run;

data work.new_rar_analysis2;
  merge work.new_rar_analysis1 work.countries2 (drop = index);
    by country;
run;

/* Transpose data */

proc transpose data=work.new_rar_analysis2 out=work.new_rar_analysis3 (rename = (_name_ = parm
col1 = ordervar));
  by country countryn;
  var aware bor_close domestic admin general health internat policies pub_gather lock school
healthsys bus_close curfews eco_meas visa emergency;
run;

proc sort data=work.new_rar_analysis3;
  by parm;
run;

proc transpose data=work.new_rar_analysis3 out=work.rar_analysist prefix=country;
  by parm;
  id countryn;
  var ordervar;
run;


ods pdf file   = "../TLF output/Google_RAR_cluster_analysis_output_excluding NPIs less than 3
weeks.pdf" style = journal bookmarklist=hide nogtitle nogfootnote;

title1 "Cluster analyis on all NPIs occurring within 3 weeks from sabilizing";

proc distance data=work.rar_analysist method=djaccard absent=0 out=distjacc;
  var anominal(country1 country2 country3 country4 country5 country6 country7 country8
country9 country10
```

```
                country11 country12 country13 country14 country15 country16 country17
country18 country19 country20
                country21 country22 country23 country24 country25 country26 country27
country28 country29 country30
                country31 country32 country33 country34 country35 country36 country37
country38 country39 country40
                country41 country42 country43 country44 country45 country46 country47
country48 country49 country50
                country51 country52 country53 country54 country55 country56 country57
country58 country59 country60
                country61 country62 country63 country64 country65 country66 country67
country68 country69 country70
                country71 country72 country73 country74 country75 country76 country77
country78 country79 country80
                country81 country82 country83 country84 country85 country86 country87
country88 country89 country90
                country91 country92 country93 country94 country95 country96 country97
country98 country99 country100
                country101 country102 country103 country104 country105 country106 country107
country108 country109 country110
                country111 country112 country113 country114 country115 country116 country117
country118 country119 country120
                country121);
  id parm;
run;


proc print data=distjacc;
  id parm;
run;

proc cluster data=distjacc method=centroid pseudo outtree=tree;
   id parm;
run;


ods pdf close;








*----------------------------------------------------------------------------------
                             GOOGLE TS DATA
-----------------------------------------------------------------------------------;


data work.new_ts_analysis;
  set adam.npi_google_ts;
    if flock ne . and adjusted_stable_start_date ne . and flock <= adjusted_stable_start_date
then lock = flock;
       else if plock ne . and adjusted_stable_start_date ne . and plock <=
adjusted_stable_start_date and lock = . then lock = plock;
     format lock;

    array create_var{*} aware bor_check bor_close prison domestic admin general health
internat policies imp_exp pub_gather pubhealth
                        plock school emergency healthsys monitor check bus_close curfews
eco_meas mil_dep flock protgear docs
                        rlock masstest test visa funeral psych human cbor_close medtest lock;
      do i = 1 to 36;
        if create_var(i) ne . and adjusted_stable_start_date - 21 <= create_var(i) <=
adjusted_stable_start_date then create_var(i) = 1;
          else create_var(i) =0;
      end;

      format aware bor_check bor_close prison domestic admin general health internat policies
imp_exp pub_gather pubhealth
                        plock school emergency healthsys monitor check bus_close curfews
eco_meas mil_dep flock protgear docs
                        rlock masstest test visa funeral psych human cbor_close medtest lock;

      no_interventions_3weeks = sum(aware, bor_check, bor_close, prison, domestic, admin,
general, health, internat, policies, imp_exp, pub_gather,
                            pubhealth, school, emergency, healthsys, monitor, check,
bus_close, curfews, eco_meas, mil_dep, lock,
                            protgear, docs, rlock, masstest, test, visa, funeral, psych,
human, cbor_close, medtest);


run;

proc sort data=work.new_ts_analysis out=work.new_ts_analysis1 (where = (country not in
("Mongolia", "Yemen", "Japan"))) nodupkey;
  by country adjusted_stable_start_date;
run;

proc means data=work.new_ts_analysis1 n mean std median mode min max;
  var no_interventions_3weeks;
  output out=stat_no_interventions_3weeks
    n = n
    mean = mean
    std = std
    median = median
    mode = mode
    min = min
    max = max;
run;


/* Calculating Chi-square results */
%let v1 = aware;
%let v2 = bor_close;
%let v3 = domestic;
%let v4 = admin;
%let v5 = general;
%let v6 = health;
%let v7 = internat;
%let v8 = policies;
%let v9 = pub_gather;
%let v10 = lock;
%let v11 = school;
%let v12 = emergency;
%let v13 = healthsys;
%let v14 = bus_close;
%let v15 = curfews;
%let v16 = eco_meas;
%let v17 = visa;

data work.chisqr_output;
  length table $256. statistic $28. ;
  table = "";
  statistic = "";
  df = .;
  value = .;
  prob = .;
run;

options spool;
%macro chisquare(indata=);

%DO i=1 %TO 16;
  %DO j=(&i+1) %TO 17;
   ods output chisq = work.stat_output;
   PROC FREQ DATA = &indata. ;
      %STR(tables  &&v&i * &&v&j / chisq OUT= out_&i._&j.; );
   RUN;

    PROC DATASETS library=work force;
      APPEND BASE=work.chisqr_output  data=work.stat_output;
    QUIT;
    RUN;
   ods output close;

  %END;
%END;

%mend;


%chisquare(indata=work.new_rar_analysis1);
```

```
data adam.TS_int_chisqr_excNPI3weeks;
  set work.chisqr_output;
     if statistic = "Chi-Square";
     if prob < 0.05 then sig_flag = "Y";
run;




/*      PERFORM CLUSTER ANALYSIS ON THE ORDER VARIABLE */
/* Determine unique countries and label them 1 to 59 */

proc sort data=work.new_ts_analysis1 out=work.countries (keep = country) nodupkey;
  by country;
run;

data work.countries1;
  set work.countries;
    index = 1;
run;

proc sort data=work.countries1;
  by index;
run;

data work.countries2;
  set work.countries1;
    by index;
      if first.index then countryn = 1;
        else countryn + 1;
run;

data work.new_ts_analysis2;
  merge work.new_ts_analysis1 work.countries2 (drop = index);
    by country;
run;

/* Transpose data */

proc transpose data=work.new_ts_analysis2 out=work.new_ts_analysis3 (rename = (_name_ = parm
col1 = ordervar));
  by country countryn;
  var aware bor_close domestic admin general health internat policies pub_gather lock school
healthsys bus_close curfews eco_meas visa emergency;
run;

proc sort data=work.new_ts_analysis3;
  by parm;
run;

proc transpose data=work.new_ts_analysis3 out=work.ts_analysist prefix=country;
  by parm;
  id countryn;
  var ordervar;
run;


ods pdf file   = "../TLF output/Google_TS_cluster_analysis_output_excluding NPIs less than 3
weeks.pdf" style = journal bookmarklist=hide nogtitle nogfootnote;

title1 "Cluster analyis on all NPIs occurring within 3 weeks from sabilizing";

proc distance data=work.ts_analysist method=djaccard absent=0 out=distjacc;
  var anominal(country1 country2 country3 country4 country5 country6 country7 country8
country9 country10
                country11 country12 country13 country14 country15 country16 country17
country18 country19 country20
                country21 country22 country23 country24 country25 country26 country27
country28 country29 country30
                country31 country32 country33 country34 country35 country36 country37
country38 country39 country40
                country41 country42 country43 country44 country45 country46 country47
country48 country49 country50
                country51 country52 country53 country54 country55 country56 country57
country58 country59 country60
```

```
                country61 country62 country63 country64 country65 country66 country67
country68 country69 country70
                country71 country72 country73 country74 country75 country76 country77
country78 country79 country80
                country81 country82 country83 country84 country85 country86 country87
country88 country89 country90
                country91 country92 country93 country94 country95 country96 country97
country98 country99 country100
                country101 country102 country103 country104 country105 country106 country107
country108 country109 country110
                country111 country112 country113 country114 country115 country116 country117
country118 country119 country120
                country121);
  id parm;
run;

proc print data=distjacc;
  id parm;
run;

proc cluster data=distjacc method=centroid pseudo outtree=tree;
   id parm;
run;


ods pdf close;



***************************************************************************************;

/* Descriptive statistics */

***************************************************************************************;



title1 "Descriptive statistics for number of unique interventions implemented";

proc means data=&outdata. n mean mode std median min max lclm uclm;
   var no_interventions;
   output out=stats_no_interventions
    n = n
    mean = mean
    mode =mode
    std = sd
    median = median
    min = min
    max = max
    lclm = lclm
    uclm = uclm;
run;


title1 "Descriptive statistics for average % of mobility left after stabilizing";

proc means data=&outdata. n mean std median min max lclm uclm;
   var &varb.;
   output out=stats_stable
    n = n
    mean = mean
    std = sd
    median = median
    min = min
    max = max
    lclm = lclm
    uclm = uclm;
run;


title1 "Descriptive statistics for average % of mobility LOST after stabilizing";

proc means data=&outdata. n mean std median min max lclm uclm;
   var &varblost.;
   output out=stats_stable_lost
    n = n
    mean = mean
    std = sd
    median = median
    min = min
    max = max
    lclm = lclm
    uclm = uclm;
run;
```

```
*************************************************************************************;
```

/* Correlation analysis */

```
*************************************************************************************;
```

```
%Macro CorAnalysis(ParaDesc=Average % mobility lost after
stabilizing,GoogleKind=RAR,AppleVar=apple_mob,GoogleVar=rar_mob,where=,n=56);
%if %length(&where)>0 %then %let where2=WHERE &where;
%else %let where2=;
Title "&ParaDesc: Google &GoogleKind vs Apple (N=&n)";
Proc Corr data=MyCor;
&Where2;
var &AppleVar;
with &GoogleVar;
run;

proc glm data=MyCor;
&Where2;
model &GoogleVar=&AppleVar;
run;

proc sgplot data=MyCor;
&Where2;
reg x=&AppleVar y=&GoogleVar / clm cli;
xaxis label="Apple";
yaxis label= "Google &Googlekind";
run;
%mend;
```