

```

1  /* Problem Statement 1 SAS Code */
2
3  /* Import data */
4  FILENAME REFFILE '/folders/myfolders/project/vehicles19.xlsx';
5
6  PROC IMPORT DATAFILE=REFFILE
7      DBMS=XLSX
8      OUT=WORK.VEHICLE;
9      GETNAMES=YES;
10 RUN;
11
12 PROC CONTENTS DATA=WORK.VEHICLE; RUN;
13
14 /* Drop unused variables */
15 data work.vehicle; set work.vehicle;
16     keep collision_id crash_date crash_time vehicle_type;
17 run;
18
19 /* Get crash weekday, crash month, crash hour */
20 data work.vehicle; set work.vehicle;
21     crash_weekday = weekday(crash_date);
22     crash_month = month(crash_date);
23     crash_hour = hour(crash_time);
24     drop crash_date crash_time;
25     rename crash_hour=crash_time;
26 run;
27
28 /* Process vehicle type */
29 /* Find crash frequency of vehicle types */
30 PROC SQL;
31 CREATE TABLE WORK.FREQUENCY
32 AS
33 SELECT COUNT(VEHICLE.COLLISION_ID)
34 AS Frequency, VEHICLE.VEHICLE_TYPE
35 FROM WORK.VEHICLE VEHICLE
36 GROUP BY VEHICLE.VEHICLE_TYPE;
37 QUIT;
38
39 /* Subset vehicle types with frequency higher than 100 */
40 /* Insert frequency into main dataset */
41 data work.vehicle2; set work.vehicle;
42 run;
43
44 PROC SQL;
45 CREATE TABLE WORK.VEHICLE
46 AS
47 SELECT VEHICLE2.COLLISION_ID, VEHICLE2.VEHICLE_TYPE, VEHICLE2.crash_weekday, VEHICLE2.crash_month, VEHICLE2.c
48 FROM WORK.VEHICLE2 VEHICLE2
49 INNER JOIN WORK.FREQUENCY FREQUENCY
50 ON
51     ( VEHICLE2.VEHICLE_TYPE = FREQUENCY.VEHICLE_TYPE ) ;
52 QUIT;
53
54 /* Subset */
55 data work.vehicle; set work.vehicle;
56     where frequency >=100;
57 run;
58
59 data work.vehicle; set work.vehicle;
60     where not missing (vehicle_type);
61 run;
62
63 /* Reclassify vehicle types */
64 data work.vehicle; set work.vehicle;
65     length classify $38;
66     if vehicle_type = "Box Truck" then classify = "Truck";
67     else if vehicle_type = "Tractor Truck Diesel" then classify = "Truck";
68     else if vehicle_type = "Flat Bed" then classify = "Truck";
69     else if vehicle_type = "Dump" then classify = "Truck";
70     else if vehicle_type = "Station Wagon/Sport Utility Vehicle" then classify = "SUV";
71     else if vehicle_type = "Garbage or Refuse" then classify = "Truck";
72     else if vehicle_type = "PK" then classify = "Pick-up Truck";
73     else if vehicle_type = "4 dr sedan" then classify = "Sedan";
74     else classify = vehicle_type;
75     drop vehicle_type;
76     rename classify = vehicle_type;
77 run;
78
79 /* Subset 10 vehicle types with highest crash frequency */
80 /* Check frequency */

```

```

81 PROC SQL;
82 CREATE TABLE WORK.FREQUENCY
83 AS
84 SELECT COUNT(VEHICLE.COLLISION_ID)
85 AS Frequency, VEHICLE.VEHICLE_TYPE
86 FROM WORK.VEHICLE VEHICLE
87 GROUP BY VEHICLE.VEHICLE_TYPE;
88 QUIT;
89
90 /* Insert frequency to main dataset */
91 data work.vehicle2; set work.vehicle;
92 run;
93
94 PROC SQL;
95 CREATE TABLE WORK.VEHICLE
96 AS
97 SELECT VEHICLE2.COLLISION_ID, VEHICLE2.VEHICLE_TYPE, VEHICLE2.crash_weekday, VEHICLE2.crash_month, VEHICLE2.c:
98 FROM WORK.VEHICLE2 VEHICLE2
99 INNER JOIN WORK.FREQUENCY FREQUENCY
100 ON
101 ( VEHICLE2.VEHICLE_TYPE = FREQUENCY.VEHICLE_TYPE ) ;
102 QUIT;
103
104 /* Subset */
105 data work.vehicle; set work.vehicle;
106 if frequency > 1000;
107 run;
108
109 PROC SQL;
110 CREATE TABLE WORK.FREQUENCY
111 AS
112 SELECT COUNT(VEHICLE.COLLISION_ID)
113 AS Frequency, VEHICLE.VEHICLE_TYPE
114 FROM WORK.VEHICLE VEHICLE
115 GROUP BY VEHICLE.VEHICLE_TYPE;
116 QUIT;
117
118 /* Plot graph */
119 /* SUV & Sedan */
120 /* Time */
121 ods graphics / reset width=6.4in height=4.8in imagemap;
122
123 proc sgplot data=WORK.VEHICLE (where=(vehicle_type="Sedan" or
124 vehicle_type="SUV"));
125 vline crash_time / group=vehicle_type curvelabel;
126 yaxis grid;
127 run;
128
129 ods graphics / reset;
130
131 /* Weekday */
132 ods graphics / reset width=6.4in height=4.8in imagemap;
133
134 proc sgplot data=WORK.VEHICLE (where=(vehicle_type="Sedan" or
135 vehicle_type="SUV"));
136 vline crash_weekday / group=vehicle_type curvelabel;
137 yaxis grid;
138 run;
139
140 ods graphics / reset;
141
142 /* Month */
143 ods graphics / reset width=6.4in height=4.8in imagemap;
144
145 proc sgplot data=WORK.VEHICLE (where=(vehicle_type="Sedan" or
146 vehicle_type="SUV"));
147 vline crash_month / group=vehicle_type curvelabel;
148 yaxis grid;
149 run;
150
151 ods graphics / reset;
152
153 /* Others */
154 /* Time */
155 ods graphics / reset width=6.4in height=4.8in imagemap;
156
157 proc sgplot data=WORK.VEHICLE (where=(vehicle_type ^= "Sedan" and
158 vehicle_type ^= "SUV"));
159 vline crash_time / group=vehicle_type curvelabel;
160 yaxis grid;
161 run;

```

```
162
163 ods graphics / reset;
164
165 /* Weekday */
166 ods graphics / reset width=6.4in height=4.8in imagemap;
167
168 proc sgplot data=WORK.VEHICLE (where=(vehicle_type ^= "Sedan" and
169     vehicle_type ^= "SUV"));
170     vline crash_weekday / group=vehicle_type curvelabel;
171     yaxis grid;
172 run;
173
174 ods graphics / reset;
175
176 /* Month */
177 ods graphics / reset width=6.4in height=4.8in imagemap;
178
179 proc sgplot data=WORK.VEHICLE (where=(vehicle_type ^= "Sedan" and
180     vehicle_type ^= "SUV"));
181     vline crash_month / group=vehicle_type curvelabel;
182     yaxis grid;
183 run;
184
185
```

```

1  /* Problem Statement 2 SAS Code*/
2
3
4  /* Pre-Processing */
5  /* import data 'crashes' */
6
7  filename MVCc
8      '/folders/myfolders/group project/Motor Vehicle Collisions - Crashes.json';
9  libname MVCc JSON fileref=MVCc;
10 libname gp '/folders/myfolders/group project';
11
12 data gp.crash; set MVCc.data;
13 run;
14
15
16
17 /* run every time open the file */
18
19 libname gp '/folders/myfolders/group project';
20 data gp.preprocess_crash ; set gp.crash;
21 run;
22
23
24 data gp.try_crash; set gp.preprocess_crash;
25 run;
26
27
28 /* drop the columns*/
29 data gp.try_crash; set gp.try_crash;
30     drop ordinal_root ordinal_data element1 element2 element3 element4 element5 element6 element7 element8
31     element12 element15 element16 element17 element18;
32 run;
33
34 /* change the form of the columns and rename */
35 data gp.try_crash; set gp.try_crash;
36     format time time. ;
37     time = input(element10,time.);
38     drop element10;
39
40     format collision_ID 8.;
41     collision_ID = input(element32,8.);
42     drop element32;
43
44     format ppl_injured ppl_killed pedes_injured pedes_killed cyc_injured
45     cyc_killed motor_injured motor_killed 4.;
46     ppl_injured = input(element19, 4.);
47     ppl_killed = input(element20, 4.);
48     pedes_injured = input(element21, 4.);
49     pedes_killed= input(element22, 4.);
50     cyc_injured= input(element23, 4.);
51     cyc_killed = input(element24, 4.);
52     motor_injured= input(element25, 4.);
53     motor_killed= input(element26, 4.);
54     drop element19 element20 element21 element22 element23 element24 element25 element26 ;
55
56     rename element9=date element11=BOROUGH element13=LATITUDE element14=LONGITUDE
57     element27=contributing_vehicle_1 element28=contributing_vehicle_2 element29=contributing_vehicle_3
58     element30=contributing_vehicle_4 element31=contributing_vehicle_5
59     element33=vehicle_type_1 element34=vehicle_type_2 element35=vehicle_type_3
60     element36=vehicle_type_4 element37=vehicle_type_5;
61 run;
62
63 /* form the date and time variables */
64 data gp.try_crash; set gp.try_crash;
65     date_year = substr(date,1,4);
66     date_month = substr(date,6,2);
67     date_day = substr(date,9,2);
68     drop date;
69     format date2 MMDDYY10.;
70     date2 = input(cats(date_month,date_day,date_year), MMDDYY10.);
71     drop date_month date_day date_year;
72     rename date2 = date;
73 run;
74
75
76 /* count the number of vehicles involved based on the contributing_vehicle */
77
78
79 data gp.try_crash_count; set gp.try_crash;
80     do;

```

```

81   if contributing_vehicle_1='' then contributing_vehicle_1_2=0;
82   else contributing_vehicle_1_2 = 1;
83   end;
84   drop contributing_vehicle_1;
85   rename contributing_vehicle_1_2 = contributing_vehicle_1;
86 run ;
87 data gp.try_crash_count; set gp.try_crash_count;
88   do;
89     if contributing_vehicle_2='' then contributing_vehicle_2_2=0;
90     else contributing_vehicle_2_2 = 1;
91     end;
92     drop contributing_vehicle_2;
93     rename contributing_vehicle_2_2 = contributing_vehicle_2;
94 run ;
95 data gp.try_crash_count; set gp.try_crash_count;
96   do;
97     if contributing_vehicle_3='' then contributing_vehicle_3_2=0;
98     else contributing_vehicle_3_2 = 1;
99     end;
100    drop contributing_vehicle_3;
101    rename contributing_vehicle_3_2 = contributing_vehicle_3;
102 run ;
103 data gp.try_crash_count; set gp.try_crash_count;
104   do;
105     if contributing_vehicle_4='' then contributing_vehicle_4_2=0;
106     else contributing_vehicle_4_2 = 1;
107     end;
108     drop contributing_vehicle_4;
109     rename contributing_vehicle_4_2 = contributing_vehicle_4;
110 run ;
111 data gp.try_crash_count; set gp.try_crash_count;
112   do;
113     if contributing_vehicle_5='' then contributing_vehicle_5_2=0;
114     else contributing_vehicle_5_2 = 1;
115     end;
116     drop contributing_vehicle_5;
117     rename contributing_vehicle_5_2 = contributing_vehicle_5;
118 run ;
119
120 data data gp.try_crash_count; set gp.try_crash_count;
121   DO i = 1 to contributing_vehicle_1;
122     vehicle_num_1 = contributing_vehicle_1 + contributing_vehicle_2 +contributing_vehicle_3+contributing_
123   end;
124   drop i contributing_vehicle_1 contributing_vehicle_2 contributing_vehicle_3 contributing_vehicle_4 contri
125 run;
126
127 /* count the number of vehicles involved based on the contributing_vehicle */
128
129 data gp.try_crash_count; set gp.try_crash_count;
130   do;
131     if vehicle_type_1='' then vehicle_type_1_2=0;
132     else vehicle_type_1_2 = 1;
133     end;
134     drop vehicle_type_1;
135     rename vehicle_type_1_2 = vehicle_type_1;
136 run ;
137 data gp.try_crash_count; set gp.try_crash_count;
138   do;
139     if vehicle_type_2='' then vehicle_type_2_2=0;
140     else vehicle_type_2_2 = 1;
141     end;
142     drop vehicle_type_2;
143     rename vehicle_type_2_2 = vehicle_type_2;
144 run ;
145 data gp.try_crash_count; set gp.try_crash_count;
146   do;
147     if vehicle_type_3='' then vehicle_type_3_2=0;
148     else vehicle_type_3_2 = 1;
149     end;
150     drop vehicle_type_3;
151     rename vehicle_type_3_2 = vehicle_type_3;
152 run ;
153 data gp.try_crash_count; set gp.try_crash_count;
154   do;
155     if vehicle_type_4='' then vehicle_type_4_2=0;
156     else vehicle_type_4_2 = 1;
157     end;
158     drop vehicle_type_4;
159     rename vehicle_type_4_2 = vehicle_type_4;
160 run ;
161 data gp.try_crash_count; set gp.try_crash_count;

```

```

162 do;
163 if vehicle_type_5='' then vehicle_type_5_2=0;
164 else vehicle_type_5_2 = 1;
165 end;
166 drop vehicle_type_5;
167 rename vehicle_type_5_2 = vehicle_type_5;
168 run ;
169
170 data gp.try_crash_count; set gp.try_crash_count;
171 DO i = 1 to vehicle_type_1;
172     vehicle_num_2 = vehicle_type_1 + vehicle_type_2 +vehicle_type_3+vehicle_type_4+vehicle_type_5;
173 end;
174 drop i vehicle_type_1 vehicle_type_2 vehicle_type_3 vehicle_type_4 vehicle_type_5;
175 run;
176
177
178 /* calculatet the maximum number of vehicles involved */
179 data gp.try_crash_count; set gp.try_crash_count;
180     vehicle_num = max(vehicle_num_1,vehicle_num_2);
181 drop vehicle_num_1 vehicle_num_2;
182 run;
183
184 proc export
185     data=GP.TRY_CRASH_COUNT
186     dbms=csv
187     outfile="/folders/myfolders/group project/20200310/crash_count.csv"
188     replace;
189 run;
190
191 /* Pre-processing Done*/
192
193
194
195
196
197
198 /* check the multiple collison in 2019*/
199 data gp.multi_collision; set gp.try_crash_count;
200     where vehicle_num > 2;
201 run;
202
203 data gp.multi_collision2019; set gp.multi_collision;
204     where year(date) = 2019;
205 run;
206
207
208
209 /* load vehicle dataset in 2019 */
210 libname gpvehi '/folders/myfolders/group project/20200310';
211
212 FILENAME REFFILE '/folders/myfolders/group project/20200310/19 data/vehicles19.xlsx';
213
214 PROC IMPORT DATAFILE=REFFILE
215     DBMS=XLSX
216     OUT=GPVEHI.VEHICLE_2019;
217     GETNAMES=YES;
218 RUN;
219
220 PROC CONTENTS DATA=GPVEHI.VEHICLE_2019; RUN;
221
222
223
224 /* do inner join and keep variables we want: cars involved in multi-vehicle-collisions*/
225 PROC SQL;
226 CREATE TABLE GPVEHI.MULTI_VEHICLE_COLLISION_2019
227 AS
228 SELECT MULTI_COLLISION2019.collision_ID, MULTI_COLLISION2019.vehicle_num, MULTI_COLLISION2019.time, MULTI_COLLISION2019.vehicle_type_1, MULTI_COLLISION2019.vehicle_type_2, MULTI_COLLISION2019.vehicle_type_3, MULTI_COLLISION2019.vehicle_type_4, MULTI_COLLISION2019.vehicle_type_5, MULTI_COLLISION2019.vehicle_num_1, MULTI_COLLISION2019.vehicle_num_2, MULTI_COLLISION2019.vehicle_num_3, MULTI_COLLISION2019.vehicle_num_4, MULTI_COLLISION2019.vehicle_num_5, MULTI_COLLISION2019.vehicle_num_6, MULTI_COLLISION2019.vehicle_num_7, MULTI_COLLISION2019.vehicle_num_8, MULTI_COLLISION2019.vehicle_num_9, MULTI_COLLISION2019.vehicle_num_10, MULTI_COLLISION2019.vehicle_num_11, MULTI_COLLISION2019.vehicle_num_12, MULTI_COLLISION2019.vehicle_num_13, MULTI_COLLISION2019.vehicle_num_14, MULTI_COLLISION2019.vehicle_num_15, MULTI_COLLISION2019.vehicle_num_16, MULTI_COLLISION2019.vehicle_num_17, MULTI_COLLISION2019.vehicle_num_18, MULTI_COLLISION2019.vehicle_num_19, MULTI_COLLISION2019.vehicle_num_20, MULTI_COLLISION2019.vehicle_num_21, MULTI_COLLISION2019.vehicle_num_22, MULTI_COLLISION2019.vehicle_num_23, MULTI_COLLISION2019.vehicle_num_24, MULTI_COLLISION2019.vehicle_num_25, MULTI_COLLISION2019.vehicle_num_26, MULTI_COLLISION2019.vehicle_num_27, MULTI_COLLISION2019.vehicle_num_28, MULTI_COLLISION2019.vehicle_num_29, MULTI_COLLISION2019.vehicle_num_30, MULTI_COLLISION2019.vehicle_num_31, MULTI_COLLISION2019.vehicle_num_32, MULTI_COLLISION2019.vehicle_num_33, MULTI_COLLISION2019.vehicle_num_34, MULTI_COLLISION2019.vehicle_num_35, MULTI_COLLISION2019.vehicle_num_36, MULTI_COLLISION2019.vehicle_num_37, MULTI_COLLISION2019.vehicle_num_38, MULTI_COLLISION2019.vehicle_num_39, MULTI_COLLISION2019.vehicle_num_40, MULTI_COLLISION2019.vehicle_num_41, MULTI_COLLISION2019.vehicle_num_42, MULTI_COLLISION2019.vehicle_num_43, MULTI_COLLISION2019.vehicle_num_44, MULTI_COLLISION2019.vehicle_num_45, MULTI_COLLISION2019.vehicle_num_46, MULTI_COLLISION2019.vehicle_num_47, MULTI_COLLISION2019.vehicle_num_48, MULTI_COLLISION2019.vehicle_num_49, MULTI_COLLISION2019.vehicle_num_50, MULTI_COLLISION2019.vehicle_num_51, MULTI_COLLISION2019.vehicle_num_52, MULTI_COLLISION2019.vehicle_num_53, MULTI_COLLISION2019.vehicle_num_54, MULTI_COLLISION2019.vehicle_num_55, MULTI_COLLISION2019.vehicle_num_56, MULTI_COLLISION2019.vehicle_num_57, MULTI_COLLISION2019.vehicle_num_58, MULTI_COLLISION2019.vehicle_num_59, MULTI_COLLISION2019.vehicle_num_60, MULTI_COLLISION2019.vehicle_num_61, MULTI_COLLISION2019.vehicle_num_62, MULTI_COLLISION2019.vehicle_num_63, MULTI_COLLISION2019.vehicle_num_64, MULTI_COLLISION2019.vehicle_num_65, MULTI_COLLISION2019.vehicle_num_66, MULTI_COLLISION2019.vehicle_num_67, MULTI_COLLISION2019.vehicle_num_68, MULTI_COLLISION2019.vehicle_num_69, MULTI_COLLISION2019.vehicle_num_70, MULTI_COLLISION2019.vehicle_num_71, MULTI_COLLISION2019.vehicle_num_72, MULTI_COLLISION2019.vehicle_num_73, MULTI_COLLISION2019.vehicle_num_74, MULTI_COLLISION2019.vehicle_num_75, MULTI_COLLISION2019.vehicle_num_76, MULTI_COLLISION2019.vehicle_num_77, MULTI_COLLISION2019.vehicle_num_78, MULTI_COLLISION2019.vehicle_num_79, MULTI_COLLISION2019.vehicle_num_80, MULTI_COLLISION2019.vehicle_num_81, MULTI_COLLISION2019.vehicle_num_82, MULTI_COLLISION2019.vehicle_num_83, MULTI_COLLISION2019.vehicle_num_84, MULTI_COLLISION2019.vehicle_num_85, MULTI_COLLISION2019.vehicle_num_86, MULTI_COLLISION2019.vehicle_num_87, MULTI_COLLISION2019.vehicle_num_88, MULTI_COLLISION2019.vehicle_num_89, MULTI_COLLISION2019.vehicle_num_90, MULTI_COLLISION2019.vehicle_num_91, MULTI_COLLISION2019.vehicle_num_92, MULTI_COLLISION2019.vehicle_num_93, MULTI_COLLISION2019.vehicle_num_94, MULTI_COLLISION2019.vehicle_num_95, MULTI_COLLISION2019.vehicle_num_96, MULTI_COLLISION2019.vehicle_num_97, MULTI_COLLISION2019.vehicle_num_98, MULTI_COLLISION2019.vehicle_num_99, MULTI_COLLISION2019.vehicle_num_100, MULTI_COLLISION2019.vehicle_num_101, MULTI_COLLISION2019.vehicle_num_102, MULTI_COLLISION2019.vehicle_num_103, MULTI_COLLISION2019.vehicle_num_104, MULTI_COLLISION2019.vehicle_num_105, MULTI_COLLISION2019.vehicle_num_106, MULTI_COLLISION2019.vehicle_num_107, MULTI_COLLISION2019.vehicle_num_108, MULTI_COLLISION2019.vehicle_num_109, MULTI_COLLISION2019.vehicle_num_110, MULTI_COLLISION2019.vehicle_num_111, MULTI_COLLISION2019.vehicle_num_112, MULTI_COLLISION2019.vehicle_num_113, MULTI_COLLISION2019.vehicle_num_114, MULTI_COLLISION2019.vehicle_num_115, MULTI_COLLISION2019.vehicle_num_116, MULTI_COLLISION2019.vehicle_num_117, MULTI_COLLISION2019.vehicle_num_118, MULTI_COLLISION2019.vehicle_num_119, MULTI_COLLISION2019.vehicle_num_120, MULTI_COLLISION2019.vehicle_num_121, MULTI_COLLISION2019.vehicle_num_122, MULTI_COLLISION2019.vehicle_num_123, MULTI_COLLISION2019.vehicle_num_124, MULTI_COLLISION2019.vehicle_num_125, MULTI_COLLISION2019.vehicle_num_126, MULTI_COLLISION2019.vehicle_num_127, MULTI_COLLISION2019.vehicle_num_128, MULTI_COLLISION2019.vehicle_num_129, MULTI_COLLISION2019.vehicle_num_130, MULTI_COLLISION2019.vehicle_num_131, MULTI_COLLISION2019.vehicle_num_132, MULTI_COLLISION2019.vehicle_num_133, MULTI_COLLISION2019.vehicle_num_134, MULTI_COLLISION2019.vehicle_num_135, MULTI_COLLISION2019.vehicle_num_136, MULTI_COLLISION2019.vehicle_num_137, MULTI_COLLISION2019.vehicle_num_138, MULTI_COLLISION2019.vehicle_num_139, MULTI_COLLISION2019.vehicle_num_140, MULTI_COLLISION2019.vehicle_num_141, MULTI_COLLISION2019.vehicle_num_142, MULTI_COLLISION2019.vehicle_num_143, MULTI_COLLISION2019.vehicle_num_144, MULTI_COLLISION2019.vehicle_num_145, MULTI_COLLISION2019.vehicle_num_146, MULTI_COLLISION2019.vehicle_num_147, MULTI_COLLISION2019.vehicle_num_148, MULTI_COLLISION2019.vehicle_num_149, MULTI_COLLISION2019.vehicle_num_150, MULTI_COLLISION2019.vehicle_num_151, MULTI_COLLISION2019.vehicle_num_152, MULTI_COLLISION2019.vehicle_num_153, MULTI_COLLISION2019.vehicle_num_154, MULTI_COLLISION2019.vehicle_num_155, MULTI_COLLISION2019.vehicle_num_156, MULTI_COLLISION2019.vehicle_num_157, MULTI_COLLISION2019.vehicle_num_158, MULTI_COLLISION2019.vehicle_num_159, MULTI_COLLISION2019.vehicle_num_160, MULTI_COLLISION2019.vehicle_num_161, MULTI_COLLISION2019.vehicle_num_162, MULTI_COLLISION2019.vehicle_num_163, MULTI_COLLISION2019.vehicle_num_164, MULTI_COLLISION2019.vehicle_num_165, MULTI_COLLISION2019.vehicle_num_166, MULTI_COLLISION2019.vehicle_num_167, MULTI_COLLISION2019.vehicle_num_168, MULTI_COLLISION2019.vehicle_num_169, MULTI_COLLISION2019.vehicle_num_170, MULTI_COLLISION2019.vehicle_num_171, MULTI_COLLISION2019.vehicle_num_172, MULTI_COLLISION2019.vehicle_num_173, MULTI_COLLISION2019.vehicle_num_174, MULTI_COLLISION2019.vehicle_num_175, MULTI_COLLISION2019.vehicle_num_176, MULTI_COLLISION2019.vehicle_num_177, MULTI_COLLISION2019.vehicle_num_178, MULTI_COLLISION2019.vehicle_num_179, MULTI_COLLISION2019.vehicle_num_180, MULTI_COLLISION2019.vehicle_num_181, MULTI_COLLISION2019.vehicle_num_182, MULTI_COLLISION2019.vehicle_num_183, MULTI_COLLISION2019.vehicle_num_184, MULTI_COLLISION2019.vehicle_num_185, MULTI_COLLISION2019.vehicle_num_186, MULTI_COLLISION2019.vehicle_num_187, MULTI_COLLISION2019.vehicle_num_188, MULTI_COLLISION2019.vehicle_num_189, MULTI_COLLISION2019.vehicle_num_190, MULTI_COLLISION2019.vehicle_num_191, MULTI_COLLISION2019.vehicle_num_192, MULTI_COLLISION2019.vehicle_num_193, MULTI_COLLISION2019.vehicle_num_194, MULTI_COLLISION2019.vehicle_num_195, MULTI_COLLISION2019.vehicle_num_196, MULTI_COLLISION2019.vehicle_num_197, MULTI_COLLISION2019.vehicle_num_198, MULTI_COLLISION2019.vehicle_num_199, MULTI_COLLISION2019.vehicle_num_200, MULTI_COLLISION2019.vehicle_num_201, MULTI_COLLISION2019.vehicle_num_202, MULTI_COLLISION2019.vehicle_num_203, MULTI_COLLISION2019.vehicle_num_204, MULTI_COLLISION2019.vehicle_num_205, MULTI_COLLISION2019.vehicle_num_206, MULTI_COLLISION2019.vehicle_num_207, MULTI_COLLISION2019.vehicle_num_208, MULTI_COLLISION2019.vehicle_num_209, MULTI_COLLISION2019.vehicle_num_210, MULTI_COLLISION2019.vehicle_num_211, MULTI_COLLISION2019.vehicle_num_212, MULTI_COLLISION2019.vehicle_num_213, MULTI_COLLISION2019.vehicle_num_214, MULTI_COLLISION2019.vehicle_num_215, MULTI_COLLISION2019.vehicle_num_216, MULTI_COLLISION2019.vehicle_num_217, MULTI_COLLISION2019.vehicle_num_218, MULTI_COLLISION2019.vehicle_num_219, MULTI_COLLISION2019.vehicle_num_220, MULTI_COLLISION2019.vehicle_num_221, MULTI_COLLISION2019.vehicle_num_222, MULTI_COLLISION2019.vehicle_num_223, MULTI_COLLISION2019.vehicle_num_224, MULTI_COLLISION2019.vehicle_num_225, MULTI_COLLISION2019.vehicle_num_226, MULTI_COLLISION2019.vehicle_num_227, MULTI_COLLISION2019.vehicle_num_228, MULTI_COLLISION2019.vehicle_num_229, MULTI_COLLISION2019.vehicle_num_230, MULTI_COLLISION2019.vehicle_num_231, MULTI_COLLISION2019.vehicle_num_232, MULTI_COLLISION2019.vehicle_num_233, MULTI_COLLISION2019.vehicle_num_234, MULTI_COLLISION2019.vehicle_num_235, MULTI_COLLISION2019.vehicle_num_236, MULTI_COLLISION2019.vehicle_num_237, MULTI_COLLISION2019.vehicle_num_238, MULTI_COLLISION2019.vehicle_num_239, MULTI_COLLISION2019.vehicle_num_240, MULTI_COLLISION2019.vehicle_num_241, MULTI_COLLISION2019.vehicle_num
```

```
243 data GPVEHI.MULTI_VEHICLE_COLLISION_2019 ; set GPVEHI.MULTI_VEHICLE_COLLISION_2019;
244     if VEHICLE_OCCUPANTS = '' THEN DELETE;
245 RUN;
246
247 proc export
248     data=GPVEHI.MULTI_VEHICLE_COLLISION_2019
249     dbms=csv
250     outfile="/folders/myfolders/group project/20200310/multi_vehicle_collision_2019.csv"
251     replace;
252 run;
253
254 /* count the number of crashes in each day in 2019 */
255 PROC SQL;
256 CREATE TABLE gpvehi.VehicleTimeLineByDay
257 AS
258 SELECT VEHICLE_2019.crash_date, COUNT(DISTINCT VEHICLE_2019.CRASH_ID)
259 AS COLLISION_NUM
260 FROM GPVEHI.VEHICLE_2019 VEHICLE_2019
261 GROUP BY VEHICLE_2019.crash_date;
262 QUIT;
263
264
265
266
267 /* all plots used in problem statement 3 in the final report are produced by excel*/
268 /* plot the vehicle in 2019 by day*/
269 ods graphics / reset width=6.4in height=4.8in imagemap;
270
271 proc sort data=GPVEHI.VEHICLETIMELINEBYDAY out=_SeriesPlotTaskData;
272     by CRASH_DATE;
273 run;
274
275 proc sgplot data=_SeriesPlotTaskData;
276     title height=14pt "Vehicles involved in crashes in 2019 by day";
277     series x=CRASH_DATE y=COLLISION_NUM /;
278     xaxis grid;
279     yaxis grid;
280 run;
281
282 ods graphics / reset;
283 title;
284
285 proc datasets library=WORK noprint;
286     delete _SeriesPlotTaskData;
287 run;
288
289
290
291 /* export it to count the vehecles in each month */
292
293 proc export
294     data=gpvehi.VehicleTimeLine
295     dbms=csv
296     outfile="/folders/myfolders/group project/20200310/multi-timeline-2019.csv"
297     replace;
298 run;
299
300 /* Used excel to adjust the date and calculate the number of crashes by month */
301
302
303
304
305 /* import the setup data and plot */
306
307 FILENAME REFFILE '/folders/myfolders/group project/20200310/plot_vehecle2019.csv';
308
309 PROC IMPORT DATAFILE=REFFILE
310     DBMS=CSV
311     OUT=GPVEHI.PlotVehicle2019;
312     GETNAMES=YES;
313 RUN;
314
315 PROC CONTENTS DATA=GPVEHI.PlotVehicle2019; RUN;
316
317
318 /* plot the vehicle in 2019 by month*/
319 ods graphics / reset width=6.4in height=4.8in imagemap;
320
321 proc sort data=GPVEHI.PLOTVEHICLE2019 out=_SeriesPlotTaskData;
322     by MMY;
323 run;
```

```

324
325 proc sgplot data=_SeriesPlotTaskData;
326     title height=14pt "Vehicles involved in crashes in 2019 by month";
327     series x=MMYY y=COLLISION_ID /;
328     xaxis grid;
329     yaxis grid label="Vehicles_involved";
330 run;
331
332 ods graphics / reset;
333 title;
334
335 proc datasets library=WORK noprint;
336     delete _SeriesPlotTaskData;
337     run;
338
339
340
341 /* plot bar chart to show different number of vehicles involved
342 in one crash in different years*/
343 data GP.TRY_CRASH_COUNT_plot; set GP.TRY_CRASH_COUNT;
344     year = year(date);
345 run;
346
347 ods graphics / reset width=6.4in height=4.8in imagemap;
348
349 proc sgplot data=GP.TRY_CRASH_COUNT_plot;
350     vbar year / group=vehicle_num groupdisplay=cluster datalabel;
351     yaxis grid;
352 run;
353
354 ods graphics / reset;
355
356
357 /* plot pie charts to show collision type in different year */
358 data GP.TRY_CRASH_COUNT_plot; set GP.TRY_CRASH_COUNT;
359     year = year(date);
360     format collision_type $8.;
361     if vehicle_num <3 then collision_type = 'Simple';
362     else collision_type = 'Multi';
363 run;
364
365     /* by changing the year to get percentage in different year*/
366 proc template;
367     define statgraph SASStudio.Pie;
368         begingraph;
369         layout region;
370         piechart category=collision_type / stat=pct;
371         endlayout;
372         endgraph;
373     end;
374 run;
375
376 ods graphics / reset width=6.4in height=4.8in imagemap;
377
378 proc sgrender template=SASStudio.Pie
379     data=GP.TRY_CRASH_COUNT_PLOT (where=(year=2019));
380 run;
381
382 ods graphics / reset;
383
384
385
386 /* plot grams in 2019 by different variables */
387 /* plot the line gram to show the change of time */
388 data GPVEHI.MULTI_VEHICLE_COLLISION_2019_p; set GPVEHI.MULTI_VEHICLE_COLLISION_2019;
389     hour = hour(time);
390 run;
391
392 ods graphics / reset width=6.4in height=4.8in imagemap;
393
394 proc sgplot data=GPVEHI.MULTI_VEHICLE_COLLISION_2019_p;
395     vline hour /;
396     yaxis grid;
397 run;
398
399 ods graphics / reset;
400
401
402 /* plot the pie chart to show the percentage of driver's gender */
403 proc template;
404     define statgraph SASStudio.Pie;

```



```
405     begingraph;
406     layout region;
407     piechart category=DRIVER_SEX / stat=pct;
408     endlayout;
409     endgraph;
410 end;
411 run;
412
413 ods graphics / reset width=6.4in height=4.8in imagemap;
414
415 -----
416 proc sgrender template=SASStudio.Pie data=GPVEHI.MULTI_VEHICLE_COLLISION_2019;
417 run;
418
419 ods graphics / reset;
420
421 /* plot the pie chart to show the percentage of travel direction */
422 proc template;
423     define statgraph SASStudio.Pie;
424         begingraph;
425         layout region;
426         piechart category=TRAVEL_DIRECTION / stat=pct;
427         endlayout;
428         endgraph;
429     end;
430 run;
431
432 ods graphics / reset width=6.4in height=4.8in imagemap;
433
434 -----
435 proc sgrender template=SASStudio.Pie data=GPVEHI.MULTI_VEHICLE_COLLISION_2019;
436 run;
437
438 ods graphics / reset;
439
440 /* plot the pie chart to show the percentage of borough */
441 proc template;
442     define statgraph SASStudio.Pie;
443         begingraph;
444         layout region;
445         piechart category=BOROUGH / stat=pct;
446         endlayout;
447         endgraph;
448     end;
449 run;
450
451 ods graphics / reset width=6.4in height=4.8in imagemap;
452
453 -----
454 proc sgrender template=SASStudio.Pie data=GPVEHI.MULTI_VEHICLE_COLLISION_2019;
```

```
1  /* Problem Statement 3 SAS Code */
2  Libname PJ '/folders/myfolders/SAS Project';
3
4
5  FILENAME REFFILE '/folders/myfolders/SAS Project/Crash.xlsx';
6
7  PROC IMPORT DATAFILE=REFFILE
8      DBMS=xlsx
9      OUT=PJ.Crash;
10     GETNAMES=YES;
11  RUN;
12
13  FILENAME REFFILE1 '/folders/myfolders/SAS Project/Direction.xlsx';
14
15  PROC IMPORT DATAFILE=REFFILE1
16      DBMS=xlsx
17      OUT=PJ.Direction;
18     GETNAMES=YES;
19  RUN;
20
21  data PJ.crash; set pj.crash;
22
23  Drop crash_time;
24
25  run;
26
27  /*sort two dataset by collision_id */
28  proc sort data=PJ.crash;
29      by collision_id;
30  run;
31  proc sort data=PJ.direction;
32      by collision_id;
33  run;
34
35
36  /* Merge dataset by collision_id */
37  data PJ.crashmerge;
38      merge PJ.crash PJ.direction;
39      by collision_id;
40  run;
41
42  /*Turn direction into numeric values */
43  proc format;
44      invaluel direction
45          "East" = 0
46          "Northeast" = 1
47          "North" = 2
48          "Northwest" = 3
49          "West" = 4
50          "Southwest" = 5
51          "South" = 6
52          "Southeast" = 7
53          "Unknown" = 8;
```

```
54 run;
55
56 data PJ.crashmerge; set PJ.crashmerge;
57
58 Direction = input(travel_direction, direction.);
59 Drop zip_code location;
60 run;
61
62 data PJ.crashmerge;
63 run;
64
65
66
67 Proc corr data=PJ.crashmerge;
68 run;
69
70
71 /* Remove missing values */
72 data PJ.crashmerge;
73 set PJ.crashmerge;
74 if compress(cats(of _all_), '')='' then delete;
75 run;
76
77
78 Libname PJ '/folders/myfolders/Project';
79
80 FILENAME REFFILE '/folders/myfolders/Project/Crashmerge2.csv';
81
82 PROC IMPORT DATAFILE=REFFILE
83     DBMS=csv
84     OUT=PJ.Crashmerge2;
85     GETNAMES=YES;
86 RUN;
87
88 data PJ.crashmerge2; set pj.crashmerge2;
89 drop var1;
90 run;
91
92 Proc corr data=PJ.crashmerge2;
93 run;
94
95
96
97 proc reg data=pj.crashmerge2;
98
99 model crash_time_1 = direction longitude latitude;
100 run;
101
102
103
```

```

/* Problem Statement 4 SAS Code */
libname gp '/folders/myshortcuts/Project_Only_new/gp';

/*Creating the table to work for the analysis. Removed several variables which wouldn't make much sense to analysis and where
PROC SQL;
CREATE TABLE gp.persons
AS
SELECT PERSONS.COLLISION_ID, PERSONS.CRASH_DATE, PERSONS.CRASH_TIME, PERSONS.PERSON_TYPE, PERSONS.PERSON_INJURY, PERSONS.PERS
FROM WORK.PERSONS PERSONS
WHERE
(
(
(
(
(
( PERSONS.COLLISION_ID IS NOT MISSING AND PERSONS.CRASH_DATE IS NOT MISSING ) AND PERSONS.CRASH_TIME IS
) AND PERSONS.PERSON_TYPE IS NOT MISSING
) AND PERSONS.PERSON_INJURY IS NOT MISSING
) AND PERSONS.PERSON_AGE IS NOT MISSING
) AND PERSONS.PED_ROLE IS NOT MISSING
) AND PERSONS.PERSON_SEX IS NOT MISSING
) ;
QUIT;

data gp.persons; set gp.persons;
run;

data gp.workset; set gp.persons;
run;

/*Injured person observations*/
PROC SQL;
CREATE TABLE GP.Injured
AS
SELECT WORKSET.PERSON_TYPE, WORKSET.PERSON_INJURY, WORKSET.PERSON_SEX
FROM GP.WORKSET WORKSET
WHERE
( WORKSET.PERSON_INJURY = 'Injured' );
QUIT;

/* Bar graph to show the split in injured by person type*/

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

proc sgplot data=GP.INJURED;
vbar PERSON_INJURY / group=PERSON_TYPE groupdisplay=cluster;
yaxis grid;
run;

ods graphics / reset;

/* Boxplot to show what time of the day is dangerous for people types*/

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

proc sgplot data=GP.TIME_OF_DAY;
vbox CRASH_TIME / category=PERSON_TYPE;
yaxis grid;
run;

ods graphics / reset;

/* Merging datasets Persons and Crashes */
PROC SQL;
CREATE TABLE GP.MERGE
AS
SELECT WORKSET.COLLISION_ID, WORKSET.PERSON_TYPE, WORKSET.PERSON_INJURY, WORKSET.PERSON_AGE, WORKSET.PERSON_SEX, CRASHES.BORO
FROM GP.WORKSET WORKSET
INNER JOIN GP.CRASHES CRASHES
ON
( WORKSET.COLLISION_ID = CRASHES.COLLISION_ID );
QUIT;

/* Finding out which borough is dangerous by person type */

```

```
ods graphics / reset width=6.4in height=4.8in imagemap;  
  
proc sgplot data=GP.MERGE;  
  vbar PERSON_TYPE / group=BOROUGH groupdisplay=cluster;  
  yaxis grid;  
run;
```

---