

DTG를 활용한 교통사고 위험구간 예측

SA1029

정유진 김유민

목차

I. 모델의 개요

II. 분석 과정

III. 분석 결과

IV. 향후 대책 제안

V. 부록(Appendix)

1. 모델의 개요

위험운전
DTG
+
정상운전
DTG

SVM
알고리즘
모델

모델 평가

교통
시설물
설치율
비교

2. 분석 과정

✓ 분석 목적

- 교통안전공단에서 제공하는 11대 위험운전행동 기준으로 위험운전행동이 많이 발생하는 도로구간 예측
- 위험운전행동이 많이 발생하는 구간은 교통사고 발생 가능성이 높음

✓ 분석 방향

- 교통사고와 위험운전행동 간 상관관계 도출
- 교통사고와 상관성이 높은 위험운전행동을 활용하여 사고 위험구간 예측

2. 분석 과정

DTG를 활용한 교통사고 위험구간 예측

① 16~17년 DTG 중 위험운전행동 데이터 추출

전체 행: 21148296 전체 칼럼: 21

⏪ ⏴ 행 201-300 ⏵ ⏩

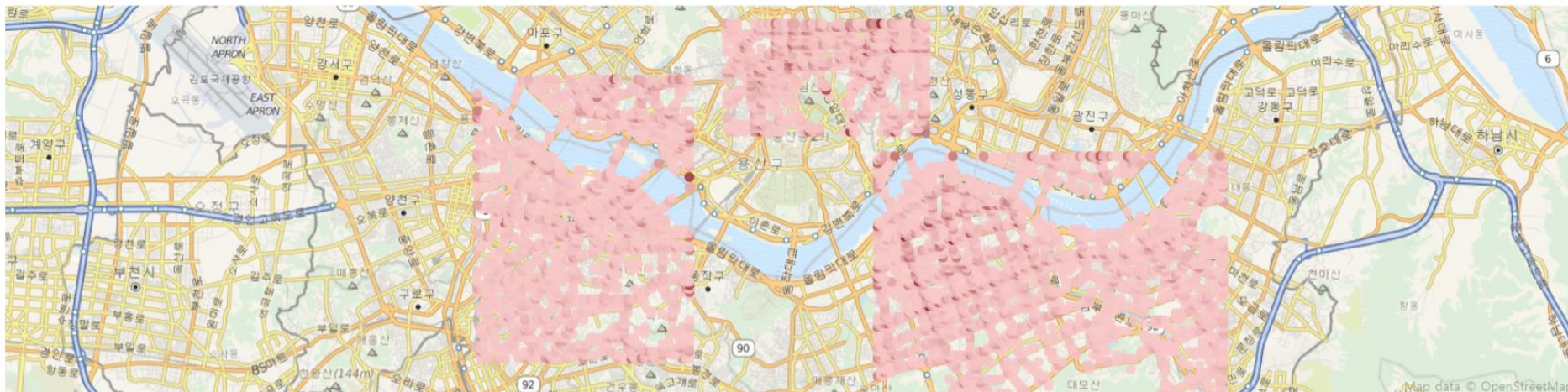
gis	Vx	Vy	company_code	time	sig_cd	rapidaccel
82	2.5	-1	XX762	160602013653	11140	1
76	2.1	1.1	XX762	160602013802	11140	1
81	2.4	-1.5	XX762	160602013847	11140	1
72	2.7	-1.4	XX762	160602013848	11140	0
79	1.4	-1.2	XX762	160602013929	11140	1
86	6.7	-2.9	XX762	160602013931	11140	0
268	2.4	-1.3	XX762	160602024118	11140	1
264	3	-2.6	XX762	160602024119	11140	0
354	2.7	-1.5	XX762	160602024422	11140	1
359	1.6	-1.3	XX762	160602024533	11140	1
265	2.2	-1	XX762	160602082248	11140	1
266	1.9	-1.2	XX762	160602082249	11140	0
257	2.9	-1.7	XX762	160602082333	11140	1
213	2.1	1	XX762	160602084136	11140	1
207	2.3	-0.9	XX762	160602084415	11140	1
7	2.1	-0.5	XX762	160602090845	11140	1
16	2.9	-1.3	XX762	160602091250	11140	1

2. 분석 과정

DTG를 활용한 교통사고 위험구간 예측

② 위경도를 이용해 교통사고와 위험운전 100m X 100m 구간화

급가속 구간 빈도 시각화



1

324

g_rapidaccel

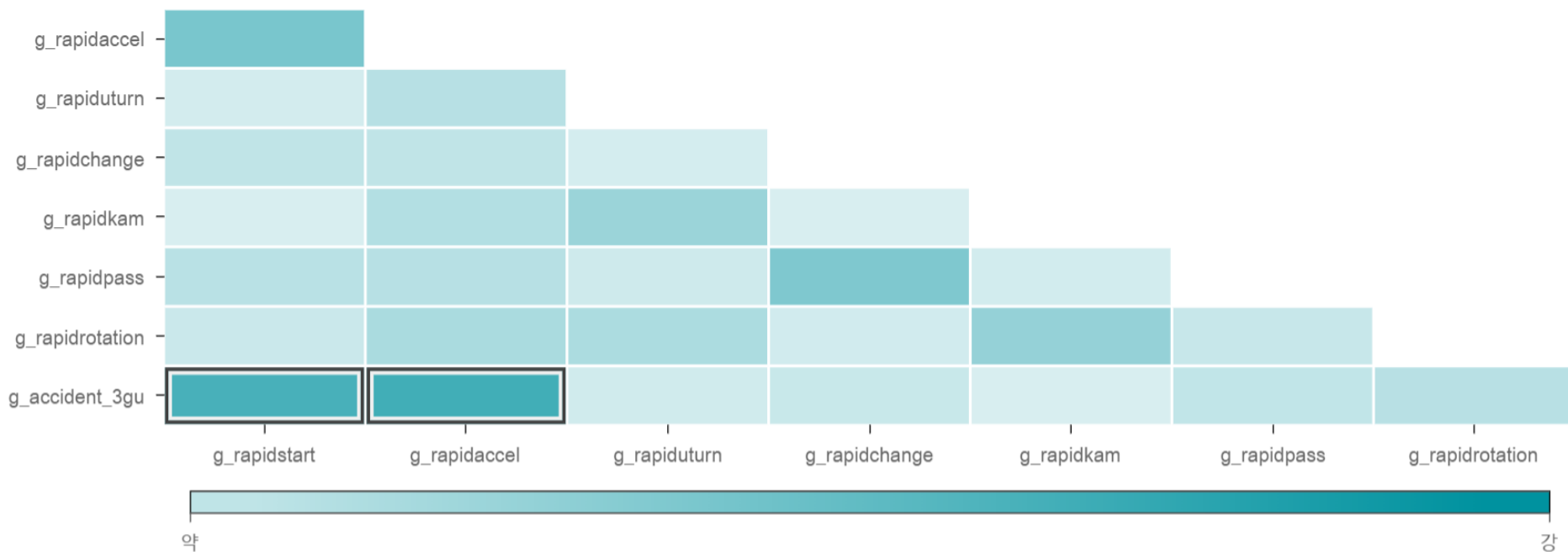
100m X 100m 격자를 기준으로 모든 분석 진행

2. 분석 과정

DTG를 활용한 교통사고 위험구간 예측

③ 교통사고와 위험운전 간 상관분석을 통해 유의미한 위험운전 도출

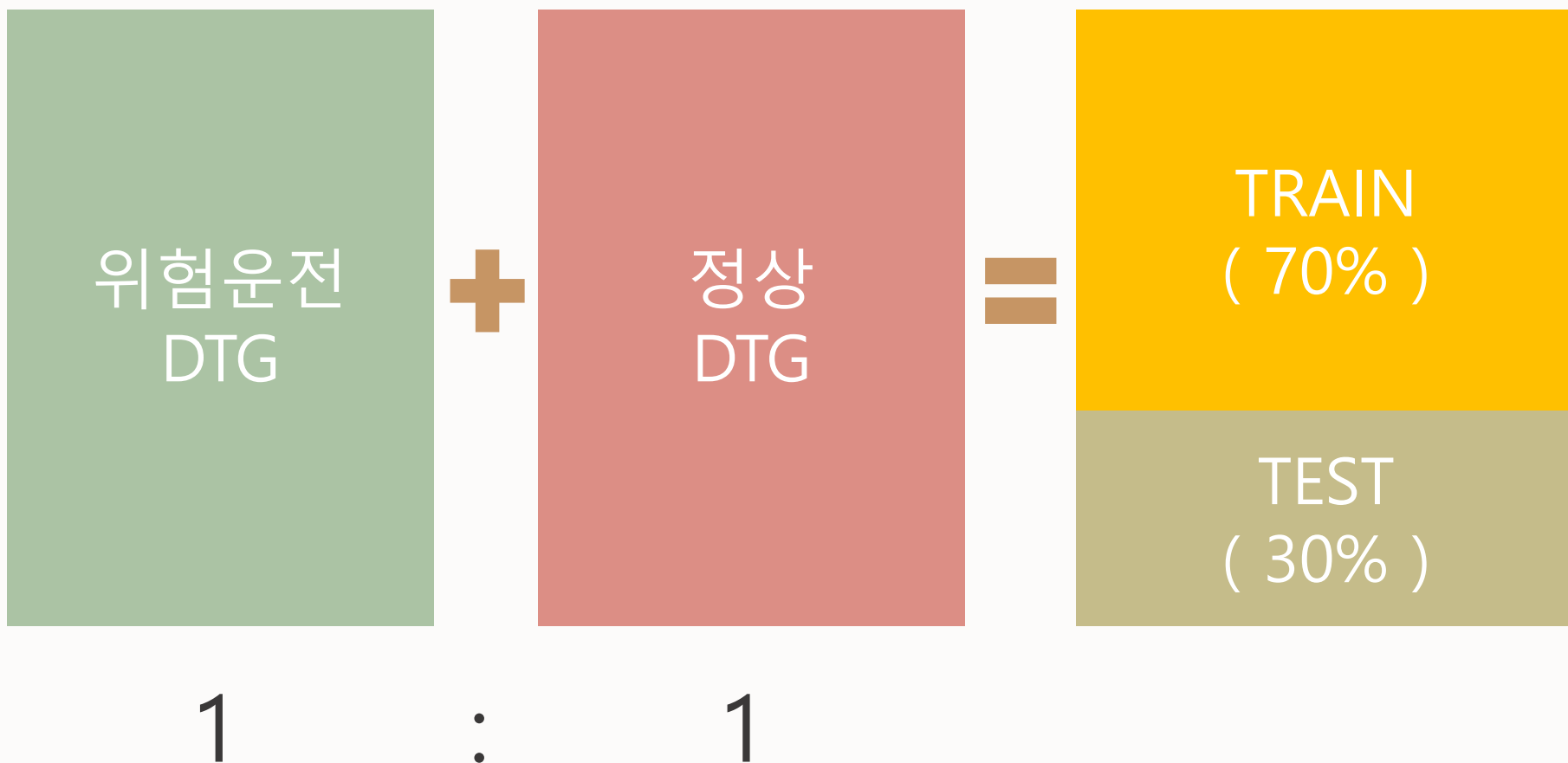
교통사고 데이터와 위험운전행동 데이터의 상관관계



교통사고는 위험운전행동 중 **급가속, 급출발**과 상관성이 높음

2. 분석 과정

④ 위험운전 분류모델 데이터 생성 (급가속, 급출발)



2. 분석 과정

DTG를 활용한 교통사고 위험구간 예측

⑤ SVM 알고리즘을 통한 모델 생성

Factorization Machine rapidaccel ASE 0.8539 사용된 관측값 207,096 사용되지 않은 관측값 224

반복 도표



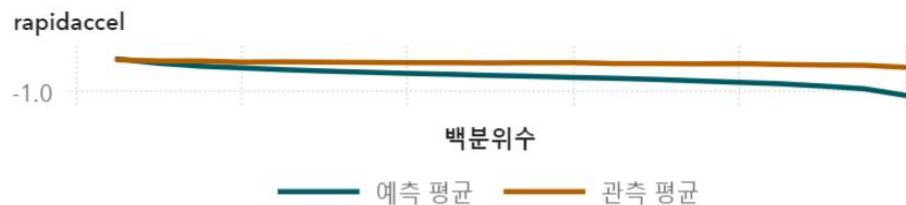
스코어된 반응



상위 추천

trip_key	순위
C-47155971917080303235200	1
C-58601383017081300000000	2
C212911106317081804074400	3
C158616897517082200000000	4

평가



3. 분석 결과

SVM 모델을 통해 잠재적 위험운전행동 데이터 도출

✓ 급가속

Misclassification Matrix			
Observed	Training Prediction		
	1	0	Total
1	103561	99	103660
0	709	102951	103660
Total	104270	103050	207320

✓ 급출발

Misclassification Matrix			
Observed	Training Prediction		
	1	0	Total
1	20592	42	20634
0	83	20551	20634
Total	20675	20593	41268

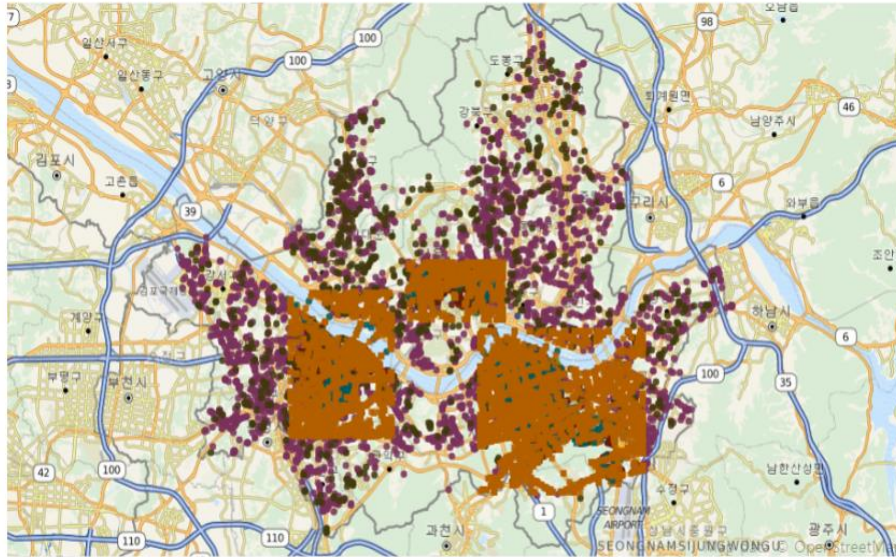
※ 잠재적 위험운동

: 실제 위험운전행동을 보이지는 않았지만 잠재적으로 위험운전행동이라고 판단되는 데이터

3. 분석 결과

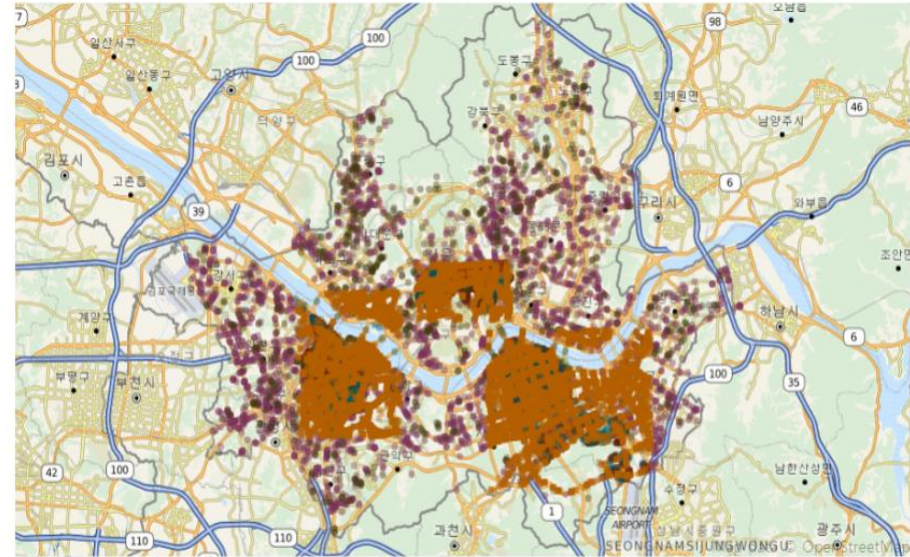
DTG를 활용한 교통사고 위험구간 예측

급출발 * 교통시설물 위치데이터



group
• bridge • busstop • carroad • cctv • child
• cross • doro • hump • living • rapidaccel

급가속 * 교통시설물 위치데이터



group
• bridge • busstop • carroad • cctv • child
• cross • doro • hump • living • rapidstart

SVM 모델을 통해 도출된
잠재적 위험운전구간 + 실제 위험운전구간과
교통시설물 설치율을 비교함

3. 분석 결과

- ✓ 실제 도로운행기록인 DTG 위험구간 데이터와의 매핑을 통해 실제 도로환경에서 발생한 위험구간 파악 및 도로시설물 등을 설치 제안하여 향후 발생 할 수 있는 교통사고를 예방
- ✓ DTG 위험구간 (10회 이상)
 - DTG 상승 급가속구간 : 5994구간
 - DTG 상승 급출발구간 : 4037구간

DTG 위험구간별 위험도 및 시설물설치율			
DTG 구분	구간수량	횡단보도	과속방지턱
상승 급가속	5994	0.54%	0.42%
상승 급출발	4037	0.05%	0.02%

3. 분석 결과

DTG를 활용한 교통사고 위험구간 예측

DTG 위험구간별 위험도 및 시설물설치율

DTG 구분	구간수량	횡단보도	과속방지턱
상습 급가속	5994	0.54%	0.42%
상습 급출발	4037	0.05%	0.02%

구간별 평균 과속방지턱 설치율

0.27

실제 과속방지턱 설치율

0.42



DTG 급과속에 해당하는 구간에서 실제 설치율이 평균 설치율보다 큰 결과를 보이며 해당 구간에서 높은 과속방지턱 설치율이 나타나고 있음

4. 향후 대책 제안

DTG 위험 구간과 매핑한 결과 과속방지턱이 설치된 구간에서
급가속이 자주 발생하며 잠재 위험도도 높은 것으로 확인됨



운전자가 과속방지턱을 지나는 순간
급가속함에 따라
위험운전행동까지 이어진 것으로 추정 가능



따라서 해당 구간에 대한
과속 카메라 설치 방안 계획 수립이 필요하며,
기존 설치된 과속방지턱의 필요성도
논의해 보아야 함

5. 부록 (Appendix)

01_초별 가속도 데이터 생성하기

```
data dtg16_6_lag;  
set sa1029.dtg16_6;  
lag=lag(speed);  
run;
```

DTG데이터의 speed 변수를 lag

```
data dtg16_6_lag;  
set work.dtg16_6_lag;  
accel=speed-lag;  
run;
```

1초 간 speed의 차를 구함

```
DATA des;  
set work.dtg16_6_lag;  
id=_n_;  
keep id accel;  
run;
```

시간 순서대로 sorting하기 위해
id를 지정

```
proc sort data = des out=tmmp;  
by descending id;  
run;
```

Id를 기준으로 파생변수 accel 을
뒤집음

5. 부록 (Appendix)

01_초별 가속도 데이터 생성하기

```
DATA tmmp;
set tmmp;
realaccel = lag(accel);
run;
```

```
proc sort data = tmmp out = tmmp;
by id;
run;
```

```
data tmmp;
set tmmp;
keep realaccel;
run;
```

```
data work.dtg16_6_lag;
merge work.dtg16_6_lag tmmp;
run;
```

```
data sa1029.dtg16_6_accel;
set work.dtg16_6_lag;
drop accel;
run;
```

뒤집은 파생변수 accel을 한 행씩 미룸

Id를 기준으로 정렬하면 초별 가속도,
즉 파생변수 realaccel이 생성됨

파생변수 realaccel과 dtg데이터를 병합

파생변수로 초별
가속도를 가진 데이터 생성

5. 부록 (Appendix)

02_위험운전 더미변수 데이터 생성하기_급가속(rapidaccel)

```
data sa1029.dtg16_6_rapidac;
set sa1029.dtg16_6_accel;
if speed>=6 and realaccel>=8 then rapidaccel=1;
else rapidaccel = 0;
run;
```

위험운전행동분석기준에 따라
급가속을 했을 경우 1,
그렇지 않을 경우 0에 해당하는 더미변수 생성

```
DATA dtg16_6_raac1;
set sa1029.dtg16_6_rapidac;
if rapidaccel = 1;
run;
```

#3초 이내 1건으로 취급하는 조건을 적용하기 위함

```
DATA dtg16_6_raac1;
set dtg16_6_raac1;
time_lag = lag(time);
time_lag2 = lag(lag(time));
if time - time_lag = 1 or time - time_lag2 = 2 or
time - time_lag = 2
then rapidaccel = 0;
run;
```

다음 행과 1초 차가 나거나,
다음 행과 2초 차가 나거나,
다다음행과 2초 차가 나는 데이터는
이전에 생성했던 더미변수를 0으로 바꿈

5. 부록 (Appendix)

02_위험운전 더미변수 데이터 생성하기_급가속 (rapidaccel)

```
DATA sa1029.dtg16_6_rapidac;  
merge sa1029.dtg16_6_rapidac dtg16_6_raac1;  
drop time_lag time_lag2;  
run;
```

3초 이내 1건 취급하는 조건을 적용한 더미변수로 데이터
변경 후 불필요한 파생변수 제거

```
data sa1029.dtg_rapidac;  
set sa1029.dtg16_6_rapidac  
sa1029.dtg16_7_rapidac sa1029.dtg16_8_rapidac  
sa1029.dtg17_6_rapidac sa1029.dtg17_7_rapidac  
sa1029.dtg17_8_rapidac;  
run;
```

위의 과정을 모두 적용한 16년 6월 ~ 17년 8월 데이터를
모두 병합함

5. 부록 (Appendix)

02_위험운전 더미변수 데이터 생성하기_급출발(rapidstart)

```
data sa1029.dtg16_6_rapidstart;  
set sa1029.dtg16_6_accel;  
if realaccel>=10 and speed<=5  
then rapidstart = 1 ;  
else rapidstart = 0 ;  
run;
```

위험운전행동분석기준에 따라
급출발을 했을 경우 1,
그렇지 않을 경우 0에 해당하는 더미변수 생성

```
data sa1029.dtg_rapidstart;  
set sa1029.dtg16_6_rapidstart  
sa1029.dtg16_7_rapidstart  
sa1029.dtg16_8_rapidstart  
sa1029.dtg17_6_rapidstart  
sa1029.dtg17_7_rapidstart  
sa1029.dtg17_8_rapidstart;  
run;
```

위의 과정을 모두 적용한 16년 6월 ~ 17년 8월 데이터를
모두 병합함

5. 부록 (Appendix)

02_위험운전 더미변수 데이터 생성하기_급감속 (rapidkam)

```
data sa1029.dtg16_6_rapidkam;
set sa1029.dtg16_6_accel;
if realaccel<=-14 and speed>=6
then rapidkam = 1 ;
else rapidkam = 0 ;
run;
```

위험운전행동분석기준에 따라
급감속을 했을 경우 1,
그렇지 않을 경우 0에 해당하는 더미변수 생성

```
DATA dtg16_6_rakam1;
set sa1029.dtg16_6_rapidkam;
if rapidkam = 1;
run;
```

#3초 이내 1건으로 취급하는 조건을 적용하기 위함

```
DATA dtg16_6_rakam2;
set dtg16_6_rakam1;
time_lag = lag(time);
time_lag2 = lag(lag(time));
if time - time_lag = 1 or time - time_lag2 = 2 or
time - time_lag = 2
then rapidkam = 0; run;
```

다음 행과 1초 차가 나거나,
다음 행과 2초 차가 나거나,
다다음행과 2초 차가 나는 데이터는
이전에 생성했던 더미변수를 0으로 바꿈

5. 부록 (Appendix)

02_위험운전 더미변수 데이터 생성하기_급감속(rapidkam)

```
DATA sa1029.dtg16_6_rapidkam;                                3초 이내 1건 취급하는 조건을 적용한 더미변수로  
merge sa1029.dtg16_6_rapidkam dtg16_6_rakam2;              데이터 변경 후 불필요한 파생변수 제거  
drop time_lag time_lag2;  
run;
```

```
data sa1029.dtg_rapidkam;                                    위의 과정을 모두 적용한 16년 6월 ~ 17년 8월 데이터를  
set sa1029.dtg16_6_rapidkam                                  모두 병합함  
sa1029.dtg16_7_rapidkam  
sa1029.dtg16_8_rapidkam  
sa1029.dtg17_6_rapidkam  
sa1029.dtg17_7_rapidkam  
sa1029.dtg17_8_rapidkam;  
run;
```

5. 부록 (Appendix)

02_위험운전 더미변수 데이터 생성하기_급정지(rapidstop)

```
data sa1029.dtg16_6_rapidstop;
set sa1029.dtg16_6_accel;
lag_accel = lag(realaccel);
if lag_accel<=-14 and speed<=5
then rapidstop = 1 ;
else rapidstop = 0 ;
run;
```

위험운전행동분석기준에 따라
급정지를 했을 경우 1,
그렇지 않을 경우 0에 해당하는 더미변수 생성

(14km/h이상 감속 후 속도가 변한 위험운전이므로
가속도를 한 행씩 미룬 파생변수 생성,
기준에 따라 필터링함)

```
data sa1029.dtg_rapidstop;
set sa1029.dtg16_6_rapidstop
sa1029.dtg16_7_rapidstop
sa1029.dtg16_8_rapidstop
sa1029.dtg17_6_rapidstop
sa1029.dtg17_7_rapidstop
sa1029.dtg17_8_rapidstop;
run;
```

위의 과정을 모두 적용한 16년 6월 ~ 17년 8월 데이터를
모두 병합함

5. 부록 (Appendix)

02_위험운전 더미변수 데이터 생성하기_급좌우회전(rapidrotation)

```
data dtg16_6_rotation;
set sa1029.dtg16_6;
gis_lag1=lag(gis);
gis_lag=lag(gis_lag1);
rotation1=gis-gis_lag1;
rotation=gis-gis_lag;
run;
```

DTG데이터의 gis 변수를 2번 lag

3초간 gis의 차를 구함

```
data sa1029.dtg16_6_rapidrota;
set dtg16_6_rotation;
if speed>=30 and 60<=rotation<=120 then
rapidrotation=1;
else if speed>=30 and -120<=rotation<=-60 then
rapidrotation=1;
else rapidrotation = 0;
run;
```

위험운전행동 분석기준에 따라
급회전을 했을 경우 1,
그렇지 않을 경우 0에 해당하는 더미변수 생성

```
DATA des;
set sa1029.dtg16_6_rapidrota;
id=_n_;
run;
```

시간 순서대로 sorting하기 위해 id를 지정

5. 부록 (Appendix)

02_위험운전 더미변수 데이터 생성하기_급좌우회전(rapidrotation)

```
DATA des;  
set des;  
keep id rotation1 rotation;  
run;
```

```
proc sort data = des out=tmp;  
by descending id;  
run;
```

Id를 기준으로 파생변수 rotation을 뒤집음

```
DATA tmp;  
set tmp;  
rotation_1 = lag(rotation1);  
rotation_2 = lag(rotation);  
rotation_3 = lag(rotation_2);  
run;
```

뒤집은 파생변수 rotation을 2행씩 미룸

```
proc sort data = tmp out = tmp;  
by id;  
run;
```


5. 부록 (Appendix)

02_위험운전 더미변수 데이터 생성하기_급좌우회전(rapidrotation)

```
data tmp;
set tmp;
keep rotation_1 rotation_3;
run;
```

Id를 기준으로 정렬하면 3초 이내 회전각,
즉 파생변수 rotation_3이 생성됨

```
data sa1029.dtg16_6_rapidrota;
merge sa1029.dtg16_6_rapidrota tmp;
run;
```

파생변수 rotation_3과 dtg데이터를 병합

```
data sa1029.dtg16_6_rotation;
set sa1029.dtg16_6_rapidrota;
drop rapidrotation rotation rotation1;
run;
```

파생변수로 3초 이내 회전각을 가진 데이터 생성

```
data sa1029.dtg16_6_rapidrota;
set sa1029.dtg16_6_rotation;
if speed>=30 and 60<=rotation_3<=120 then
rapidrotation=1;
else if speed>=30 and -120<=rotation_3<=-60 then
rapidrotation=1;
else rapidrotation = 0;
run;
```

위험운전행동 분석기준에 따라
급회전을 했을 경우 1,
그렇지 않을 경우 0에 해당하는 더미변수 생성

5. 부록 (Appendix)

02_위험운전 더미변수 데이터 생성하기_급좌우회전(rapidrotation)

```
data sa1029.dtg_rapidrotation;  
set sa1029.dtg16_6_rapidrota  
sa1029.dtg16_7_rapidrota  
sa1029.dtg16_8_rapidrota  
sa1029.dtg17_6_rapidrota  
sa1029.dtg17_7_rapidrota  
sa1029.dtg17_8_rapidrota;  
run;
```

위의 과정을 모두 적용한 16년 6월 ~ 17년 8월 데이터를
모두 병합함

5. 부록 (Appendix)

02_위험운전 더미변수 데이터 생성하기_급U턴(rapiduturn)

```
data dtg16_6_turn;  
set sa1029.dtg16_6;  
lag1=lag(gis);  
lag2=lag(lag1);  
lag3=lag(lag2);  
lag4=lag(lag3);  
gis_lag=lag(lag4);  
turn=gis-gis_lag;  
run;
```

DTG데이터의 gis변수를 5번 lag

5초 간 회전각의 차를 구함

```
data sa1029.dtg16_6_uturn;  
set dtg16_6_turn;  
if speed>=25 and 160<=turn<=180 then uturn=1;  
else if speed>=25 and -180<=turn<=-160 then  
uturn=1;  
else uturn = 0;  
run;
```

위험운전행동 분석기준에 따라
급유턴을 했을 경우 1,
그렇지 않을 경우 0에 해당하는 더미변수 생성

5. 부록 (Appendix)

02_위험운전 더미변수 데이터 생성하기_급U턴(rapiduturn)

```
DATA des;
set sa1029.dtg16_6_uturn;
id=_n_;
run;
```

시간 순서대로 sorting하기 위해
id를 지정

```
DATA des;
set des;
keep id turn;
run;
```

id를 기준으로 파생변수 turn을 뒤집음

```
proc sort data = des out=tmp;
by descending id;
run;
```

뒤집은 파생변수 trun을 5행씩 미룸

```
DATA tmp;
set tmp;
turn_1 = lag(turn);
turn_2 = lag(turn_1);
turn_3 = lag(turn_2);
turn_4 = lag(turn_3);
turn_5 = lag(turn_4);
run;
```

5. 부록 (Appendix)

02_위험운전 더미변수 데이터 생성하기_급U턴(rapiduturn)

```
proc sort data = tmmp out = tmmp;  
by id;  
run;
```

```
data tmmp;  
set tmmp;  
keep turn_5;  
run;
```

```
data sa1029.dtg16_6_urn;  
merge sa1029.dtg16_6_urn tmmp;  
run;
```

Id를 기준으로 정렬하면 6초 이내 회전각,
즉 파생변수 turn_5가 생성됨

파생변수 turn_5와 dtg데이터를 병합

5. 부록 (Appendix)

02_위험운전 더미변수 데이터 생성하기_급U턴(rapiduturn)

```
Data sa1029.dtg16_6_urn;  
set dtg16_6_turn;  
if speed>=25 and 160<=turn_5<=180 then urn=1;  
else if speed>=25 and -180<=turn_5<=-160 then  
urn=1;  
else urn = 0;  
run;
```

위험운전행동 분석기준에 따라
급유턴을 했을 경우 1,
그렇지 않을 경우 0에 해당하는 더미변수 생성

```
data sa1029.dtg_rapiduturn;  
set sa1029.dtg16_6_urn sa1029.dtg16_7_urn  
sa1029.dtg16_8_urn sa1029.dtg17_6_urn  
sa1029.dtg17_7_urn sa1029.dtg17_8_urn;  
run;
```

위의 과정을 모두 적용한 16년 6월 ~ 17년 8월 데이터를
모두 병합함

5. 부록 (Appendix)

02_위험운전 더미변수 데이터 생성하기_급앞지르기(rapidpass)

```
DATA sa1029.dtg16_6_rapidpass;
set sa1029.dtg16_6_accel;
gis_lag = lag(gis);
gis_lag_by5 = lag(lag(lag(lag(gis))));
if ABS(gis_lag - gis) >= 10 and speed >= 30 and
ABS(gis_lag_by5 - gis) <= 2
and ABS(realaccel) >= 3 then rapidpass = 1 ;
else rapidpass = 0;
run;
```

위험운전행동분석기준에 따라
급앞지르기를 했을 경우 1,
그렇지 않을 경우 0에 해당하는 더미변수 생성

(절댓값 ABS 함수를 이용해 좌/우측, 가/감속 을
판단함)

```
data sa1029.dtg_rapidpass;
set sa1029.dtg16_6_rapidpass
sa1029.dtg16_7_rapidpass
sa1029.dtg16_8_rapidpass
sa1029.dtg17_6_rapidpass
sa1029.dtg17_7_rapidpass
sa1029.dtg17_8_rapidpass;
run;
```

위의 과정을 모두 적용한 16년 6월 ~ 17년 8월 데이터를
모두 병합함

5. 부록 (Appendix)

02_위험운전 더미변수 데이터 생성하기_급진로변경(rapidchange)

```
DATA sa1029.dtg16_6_rapidchange;
set sa1029.dtg16_6_accel;
gis_lag = lag(gis);
gis_lag_by5 = lag(lag(lag(lag(gis))));
if ABS(gis_lag - gis) >= 10 and speed >= 30 and
ABS(gis_lag_by5 - gis) <= 2
and ABS(realaccel) <= 2 then rapidchange = 1 ;
else rapidchange = 0;
run;
```

위험운전행동분석기준에 따라
급진로변경을 했을 경우 1,
그렇지 않을 경우 0에 해당하는 더미변수 생성

(절댓값 ABS 함수를 이용해 좌/우측, 가/감속 을
판단함)

```
data sa1029.dtg_rapidchange;
set sa1029.dtg16_6_rapidchange
sa1029.dtg16_7_rapidchange
sa1029.dtg16_8_rapidchange
sa1029.dtg17_6_rapidchange
sa1029.dtg17_7_rapidchange
sa1029.dtg17_8_rapidchange;
run;
```

위의 과정을 모두 적용한 16년 6월 ~ 17년 8월 데이터를
모두 병합함

5. 부록 (Appendix)

03_100m X 100m 구간별 위험운전행동과 교통사고 데이터 생성하기

```
data grid_dtg_rapidstop;
```

더미변수 rapidstart=1인 데이터만 추출

```
set sa1029.dtg_rapidstop;
```

```
if rapidstop=1;
```

```
run;
```

```
data grid_dtg_rapidstop;
```

```
set grid_dtg_rapidstop;
```

```
spacingC=100*(1/110000);
```

100m를 위도로 표현

```
spacingR=100*(1/88740);
```

100m를 경도로 표현

```
latmin=37.4249;
```

서울시 최소 위도

```
latmax=37.704;
```

서울시 최대 위도

```
lonmin=126.76;
```

서울시 최소 경도

```
lonmax=127.1855;
```

서울시 최대 경도

```
c=(latmax-latmin)/spacingC;
```

위도 셀 개수(세로)

```
r=(lonmax-lonmin)/spacingR;
```

경도 셀 개수(가로)

```
c=ceil(c);
```

```
r=ceil(r);
```

```
c1=(lat-latmin)/spacingC;
```

Dtg 위도의 셀 위치

```
r1=(lon-lonmin)/spacingR;
```

Dtg 경도의 셀 위치

```
c1=ceil(c1);
```

```
r1=ceil(r1);
```

```
g= r1 + r*(c1-1);
```

전체 셀 위치 지정

```
run;
```

5. 부록 (Appendix)

03_100m X 100m 구간별 위험운전행동과 교통사고 데이터 생성하기

```
proc freq data=grid_dtg_rapidstop noprint;  
table g / nocol norow nopercnt out=  
sa1029.freq_dtg_rapidstop(drop=percent) sparse;  
run;
```

각 셀의 rapidstart의 빈도를 구함

```
data sa1029.freq_dtg_rapidstop;  
set sa1029.freq_dtg_rapidstop;  
g_rapidstop=count;  
drop count;  
run;
```

빈도를 g_rapidstop으로 변경

위와 같은 과정을 각각의 위험운전과
교통사고 데이터에 적용함

5. 부록 (Appendix)

04_교통사고와 위험운전행동 간 상관분석

```
data sa1029.dtg_grid;  
merge sa1029.freq_accident_3gu  
sa1029.freq_dtg_rapidac  
sa1029.freq_dtg_rapidrotation  
sa1029.freq_dtg_rapidstart  
sa1029.freq_dtg_rapidpass  
sa1029.freq_dtg_rapidstop  
sa1029.freq_dtg_rapidkam  
sa1029.freq_dtg_rapiduturn  
sa1029.freq_dtg_rapidchange;  
by g;  
run;
```

```
proc corr data=sa1029.dtg_grid  
out=sa1029.corr_dtg_grid;  
var g_accident_3gu g_rapidaccel g_rapidstart  
g_rapidkam g_rapidstop  
g_rapidrotation g_rapiduturn g_rapidpass  
g_rapidchange;  
run;
```

상관분석을 위한 데이터 생성

(각각의 구간별 위험운전 데이터와
구간별 교통사고 데이터를 병합)

위에서 생성한 데이터로 상관분석 진행

5. 부록 (Appendix)

05_위험운전행동 분류모델 데이터 생성

```
data sampling.dtg_rapidac1;  
set sa1029.dtg_rapidac;  
if rapidaccel=1;  
run;
```

급가속 더미변수 1인 데이터 필터링해
새로운 데이터셋 생성

```
data sampling.dtg_rapidac0;  
set sa1029.dtg_rapidac;  
if rapidaccel=0;  
run;
```

급가속 더미변수 0인 데이터 필터링해
새로운 데이터셋 생성

```
proc surveyselect  
data=sampling.dtg_rapidac0  
method= srs  
n=103660  
out=sampling.rapidacsam;  
run;
```

도출된 위험운전행동 DTG 데이터 개수와
동일하게 정상 DTG 데이터 랜덤 샘플링

5. 부록 (Appendix)

05_위험운전행동 분류모델 데이터 생성

```
proc sort data= sampling.rapidacsam;  
by time;  
run;
```

```
proc sort data= sampling.dtg_rapidac1;  
by time;  
run;
```

```
data sampling.rapidac;  
set sampling.dtg_rapidac1  
sampling.rapidacsam;  
by time;  
run;
```

Time 변수를 기준으로 병합하기 위해 sorting

랜덤 샘플링한 데이터와
위험운전행동 DTG 데이터를 병합한
데이터셋 생성

감사합니다

SA1029
정유진 김유민