로지스틱회귀분석

```
library(magrittr)
```

```
## Warning: package 'magrittr' was built under R version 3.4.4
```

```
library(maps)
```

```
## Warning: package 'maps' was built under R version 3.4.4
```

```
library(ggmap)
```

```
## Warning: package 'ggmap' was built under R version 3.4.4
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'ggmap'
```

```
## The following object is masked from 'package:magrittr':
##
##     inset
```

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 3.4.4
```

```
## corrplot 0.84 loaded
```

```
library(car)
```

```
## Warning: package 'car' was built under R version 3.4.4
```

```
## Loading required package: carData
```

```
## Warning: package 'carData' was built under R version 3.4.4
```

```
library(pROC)
```

```
## Warning: package 'pROC' was built under R version 3.4.4
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

데이터 읽기

```
df <- read.csv('1_df.csv',header=T) # 탐색용 데이터
df_var <- read.csv('final_df.csv',header=T) #분석용 데이터
```

분석용 데이터는 총 61개의 독립변수로 이루워진 테이블임. 모두 표준화음([0-1]변환)이 되어 있음 탐색용 데이터는 단속 지점 별 중복 값을

제거한 형태임.

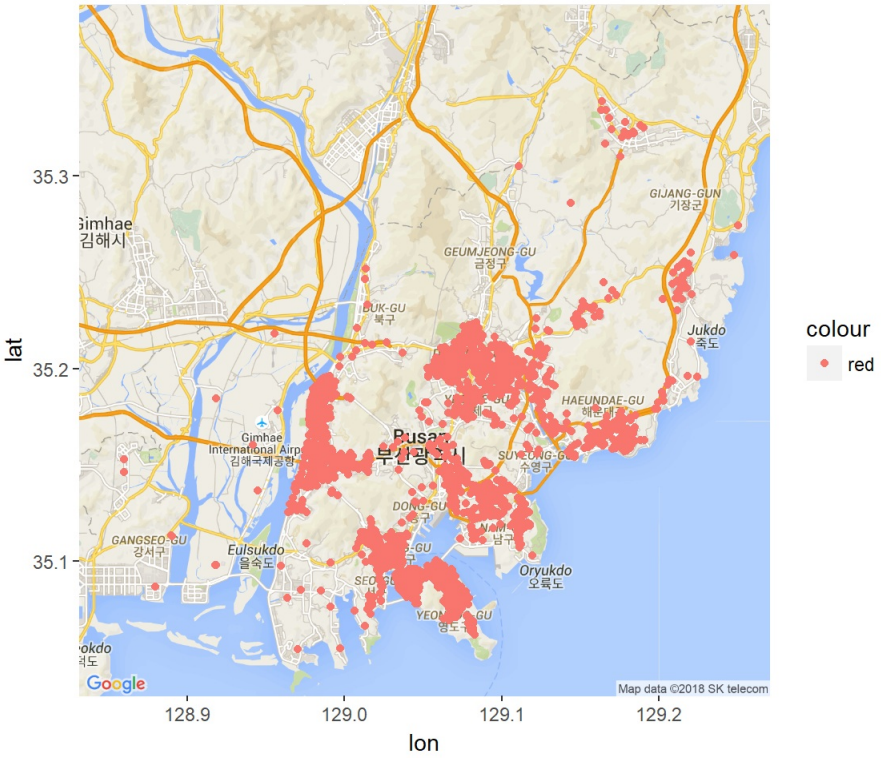
데이터 탐색

불법 주 정차 단속 현황을 지도위에 맵핑

```
busan_lat = 35.209925947222
busan_lon = 129.05055511923
get_googlemap(center=c(busan_lon,busan_lat),zoom=11,
              maptype="roadmap") %>%ggmap()+
              geom_point(aes(x=longitude,y=latitude,colour='red'),df)
```

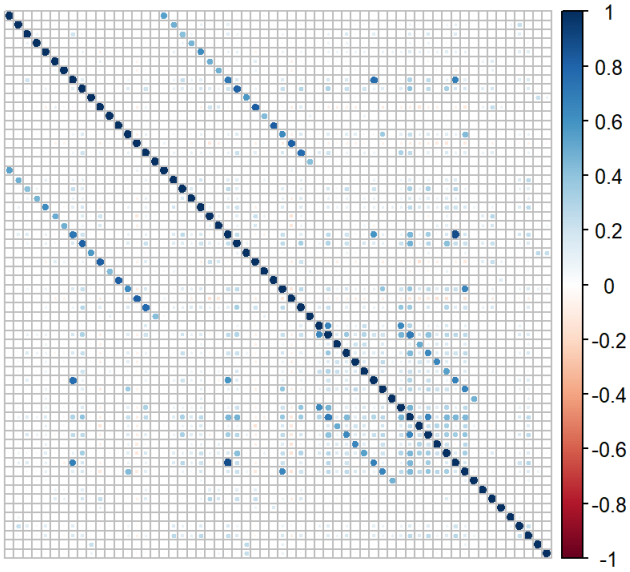
```
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=35.209926,129.050555&zoom=11&size=640x640&scale=2&maptype=roadmap&sensor=false
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```



제공 받은 구/군을 보았을 때, 거의 모든 지역이 단속된 이력이 있음.

```
corrplot(cor(df_var[,2:61]), tl.col = "white")
```



61개의 변수의 상관계수를 시각해 보았을 때, 상관 계수 0.7 이상의 값들이 존재함을 알 수 있음 다중공선성이 예상 됨.

```
#7:3 비율로 분석 데이터를 나누어 회귀 분석 진행
all_lm <- lm(y~., df_var)
vif(all_lm)
```

```
##                       GVIF Df GVIF^(1/(2*Df))
## gungu             4.267009  8        1.094921
## building_01_1     1.442333  1        1.200972
## building_07_1     1.430636  1        1.196092
## building_13_1     1.279263  1        1.131045
## building_02_1     1.318167  1        1.148114
## building_08_1     1.713353  1        1.308951
## building_03_1     1.359517  1        1.165983
## building_09_1     1.354031  1        1.163628
## building_15_1     3.621275  1        1.902965
## building_04_1     3.338856  1        1.827254
## building_10_1     1.545899  1        1.243342
## building_16_1     3.330965  1        1.825093
## building_05_1     1.267256  1        1.125725
## building_11_1     3.387594  1        1.840542
## building_17_1     1.809878  1        1.345317
## building_06_1     3.044374  1        1.744813
## building_12_1     2.741547  1        1.655762
## building_18_1     1.236358  1        1.111917
## building_01_2     1.450814  1        1.204497
## building_07_2     1.500577  1        1.224980
## building_13_2     1.446201  1        1.202581
## building_02_2     1.385122  1        1.176912
## building_08_2     2.036018  1        1.426891
## building_03_2     1.601513  1        1.265509
## building_09_2     1.470138  1        1.212492
## building_15_2     6.161743  1        2.482286
## building_04_2     4.246889  1        2.060798
## building_10_2     1.719303  1        1.311222
## building_16_2     3.710143  1        1.926173
## building_05_2     1.324757  1        1.150981
## building_11_2     3.506373  1        1.872531
## building_17_2     2.738832  1        1.654942
## building_06_2     3.338147  1        1.827060
## building_12_2     3.531090  1        1.879119
## building_18_2     1.270577  1        1.127199
## sosang_D_1        3.495134  1        1.869528
## sosang_Q_1        4.595013  1        2.143598
## sosang_L_1        1.472499  1        1.213466
## sosang_N_1        1.905035  1        1.380230
## sosang_R_1        2.008627  1        1.417260
## sosang_F_1        1.627978  1        1.275922
## sosang_O_1        3.066020  1        1.751006
## sosang_S_1        2.043261  1        1.429427
## sosang_P_1        1.343042  1        1.158897
## sosang_D_2        2.639783  1        1.624741
## sosang_Q_2        5.526711  1        2.350896
## sosang_L_2        1.713180  1        1.308885
## sosang_N_2        2.908684  1        1.705486
## sosang_R_2        2.142704  1        1.463798
## sosang_F_2        2.243138  1        1.497711
## sosang_O_2        6.155393  1        2.481006
## sosang_S_2        3.251250  1        1.803122
## sosang_P_2        1.342027  1        1.158459
## cctv_count_1      1.114084  1        1.055502
## cctv_count_2      1.388858  1        1.178498
## prior_parking_1   1.116411  1        1.056604
## prior_parking_2   1.022988  1        1.011429
## private_parking_1 1.206825  1        1.098556
## private_parking_2 1.400741  1        1.183529
## public_parking_1  1.128509  1        1.062313
## public_parking_2  1.117678  1        1.057203
```

GVIF^(1/(2*Df)) > 2인 값들이 있음으로 다중공선성이 있음. 다중공선성 제거와 유의한 계수 선별을 위해 stepwise 작업 필요

```
steped_df <- step(all_lm,direction = 'both')
```

```
## stepwise 과정이 너무 길어 html 문서에서 일부 과정을 뺐습니다.
## Step:  AIC=-5424.63
## y ~ gungu + building_13_1 + building_10_1 + building_16_1 + building_04_2 +
##     building_10_2 + building_06_2 + building_12_2 + sosang_Q_1 +
##     sosang_D_2 + sosang_F_2 + sosang_S_2 + cctv_count_1 + public_parking_2
##
##                      Df Sum of Sq    RSS     AIC
## <none>                            407.67 -5424.6
## - sosang_S_2          1     0.304 407.98 -5424.5
## - sosang_D_2          1     0.334 408.01 -5424.3
## + building_12_1       1     0.224 407.45 -5424.2
## + sosang_O_1          1     0.205 407.47 -5424.0
## + building_11_2       1     0.204 407.47 -5424.0
## + building_03_1       1     0.159 407.51 -5423.7
## + building_05_2       1     0.157 407.52 -5423.7
## + building_08_1       1     0.139 407.53 -5423.6
## + building_09_2       1     0.129 407.54 -5423.5
## + sosang_N_1          1     0.112 407.56 -5423.4
## + building_01_2       1     0.110 407.56 -5423.4
## + sosang_P_1          1     0.109 407.56 -5423.4
## + building_02_1       1     0.089 407.58 -5423.2
## + sosang_L_2          1     0.085 407.59 -5423.2
## + sosang_F_1          1     0.083 407.59 -5423.2
## + sosang_D_1          1     0.083 407.59 -5423.2
## + building_07_1       1     0.072 407.60 -5423.1
## + sosang_Q_2          1     0.071 407.60 -5423.1
## + building_15_1       1     0.053 407.62 -5423.0
## + building_09_1       1     0.053 407.62 -5423.0
## + building_11_1       1     0.049 407.62 -5423.0
## + private_parking_1   1     0.044 407.63 -5422.9
## + building_03_2       1     0.039 407.63 -5422.9
## + sosang_R_2          1     0.034 407.64 -5422.9
## + sosang_L_1          1     0.025 407.65 -5422.8
## + building_17_2       1     0.023 407.65 -5422.8
## + sosang_O_2          1     0.020 407.65 -5422.8
## + building_01_1       1     0.018 407.66 -5422.8
## + sosang_R_1          1     0.016 407.66 -5422.7
## + prior_parking_2     1     0.014 407.66 -5422.7
## + building_07_2       1     0.011 407.66 -5422.7
## + building_16_2       1     0.009 407.66 -5422.7
## + building_06_1       1     0.008 407.67 -5422.7
## + public_parking_1    1     0.007 407.67 -5422.7
## + building_15_2       1     0.006 407.67 -5422.7
## + cctv_count_2        1     0.006 407.67 -5422.7
## + sosang_N_2          1     0.006 407.67 -5422.7
## + sosang_P_2          1     0.003 407.67 -5422.7
## + building_17_1       1     0.003 407.67 -5422.6
## + sosang_S_1          1     0.002 407.67 -5422.6
## + building_05_1       1     0.001 407.67 -5422.6
## + building_08_2       1     0.001 407.67 -5422.6
## + building_18_1       1     0.001 407.67 -5422.6
## + building_04_1       1     0.001 407.67 -5422.6
## + building_13_2       1     0.000 407.67 -5422.6
## + building_18_2       1     0.000 407.67 -5422.6
## + building_02_2       1     0.000 407.67 -5422.6
## + prior_parking_1     1     0.000 407.67 -5422.6
## + private_parking_2   1     0.000 407.67 -5422.6
## - public_parking_2    1     0.602 408.28 -5422.5
## - building_10_2       1     0.629 408.30 -5422.3
## - building_12_2       1     0.641 408.31 -5422.2
## - cctv_count_1        1     0.678 408.35 -5421.9
## - sosang_F_2          1     0.710 408.38 -5421.7
## - building_04_2       1     0.731 408.40 -5421.6
## - building_13_1       1     0.976 408.65 -5419.9
## - building_06_2       1     1.193 408.87 -5418.4
## - sosang_Q_1          1     1.529 409.20 -5416.1
## - building_10_1       1     1.812 409.48 -5414.1
## - building_16_1       1     2.248 409.92 -5411.1
## - gungu               8   100.574 508.25 -4817.7
```

```
summary(steped_df)
```

```
##
## Call:
## lm(formula = y ~ gungu + building_13_1 + building_10_1 + building_16_1 +
##     building_04_2 + building_10_2 + building_06_2 + building_12_2 +
##     sosang_Q_1 + sosang_D_2 + sosang_F_2 + sosang_S_2 + cctv_count_1 +
##     public_parking_2, data = df_var)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.59706 -0.24411 -0.14492 -0.00456  0.99483
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.56421    0.05012  11.256  < 2e-16 ***
## gungu남구         -0.26016    0.05457  -4.768 1.96e-06 ***
## gungu동래구       -0.25519    0.05453  -4.680 3.01e-06 ***
## gungu사상구       -0.29058    0.05425  -5.357 9.17e-08 ***
## gungu서구         -0.32908    0.05956  -5.525 3.59e-08 ***
## gungu연제구       -0.53900    0.06639  -8.119 6.99e-16 ***
## gungu영도구       -0.40464    0.05165  -7.835 6.61e-15 ***
## gungu중구          0.45393    0.06133   7.402 1.77e-13 ***
## gungu해운대구     -0.27658    0.05599  -4.940 8.28e-07 ***
## building_13_1     0.37860    0.14613   2.591 0.009624 **
## building_10_1     0.49960    0.14156   3.529 0.000423 ***
## building_16_1    -0.28924    0.07357  -3.931 8.65e-05 ***
## building_04_2    -0.26510    0.11828  -2.241 0.025081 *
## building_10_2    -0.17697    0.08511  -2.079 0.037682 *
## building_06_2    -0.15211    0.05311  -2.864 0.004215 **
## building_12_2     0.24534    0.11689   2.099 0.035922 *
## sosang_Q_1        0.62147    0.19170   3.242 0.001201 **
## sosang_D_2       -0.28461    0.18790  -1.515 0.129974
## sosang_F_2        0.22998    0.10412   2.209 0.027271 *
## sosang_S_2       -0.11103    0.07686  -1.445 0.148670
## cctv_count_1      0.08174    0.03787   2.158 0.030984 *
## public_parking_2 -0.33216    0.16322  -2.035 0.041944 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3814 on 2803 degrees of freedom
## Multiple R-squared:  0.2302, Adjusted R-squared:  0.2244
## F-statistic: 39.91 on 21 and 2803 DF,  p-value: < 2.2e-16
```

```
vif(steped_df)
```

```
##                      GVIF Df GVIF^(1/(2*Df))
## gungu            2.063316  8        1.046310
## building_13_1    1.018989  1        1.009450
## building_10_1    1.509720  1        1.228707
## building_16_1    1.158828  1        1.076489
## building_04_2    1.784587  1        1.335884
## building_10_2    1.624331  1        1.274492
## building_06_2    1.199778  1        1.095344
## building_12_2    1.516315  1        1.231387
## sosang_Q_1       1.431627  1        1.196506
## sosang_D_2       1.519071  1        1.232506
## sosang_F_2       1.398651  1        1.182646
## sosang_S_2       1.247540  1        1.116933
## cctv_count_1     1.055907  1        1.027573
## public_parking_2 1.084128  1        1.041215
```

다중공선성이 제거 된 것을 확인 가능함.

유의한 변수를 이용하여 로지스틱 회귀분석

```
steped_glm <- glm(formula = y ~ gungu + building_13_1 + building_10_1 + building_16_1 +
    building_04_2 + building_06_2 + building_12_2 +
    sosang_Q_1 + sosang_D_2 + sosang_F_2 + sosang_S_2 + cctv_count_1 +
    public_parking_2, family='binomial', data = df_var)
vif(steped_glm)
```

```
##                     GVIF Df GVIF^(1/(2*Df))
## gungu           1.694673  8        1.033518
## building_13_1   1.023258  1        1.011562
## building_10_1   1.028025  1        1.013916
## building_16_1   1.179194  1        1.085907
## building_04_2   1.611076  1        1.269282
## building_06_2   1.202067  1        1.096388
## building_12_2   1.511396  1        1.229388
## sosang_Q_1      1.988296  1        1.410070
## sosang_D_2      2.065543  1        1.437200
## sosang_F_2      1.342528  1        1.158675
## sosang_S_2      1.224428  1        1.106539
## cctv_count_1    1.030240  1        1.015007
## public_parking_2 1.036424  1        1.018049
```

다중 공선성이 제거된 것을 확인

```
summary(steped_glm)
```

```
##
## Call:
## glm(formula = y ~ gungu + building_13_1 + building_10_1 + building_16_1 +
##     building_04_2 + building_06_2 + building_12_2 + sosang_Q_1 +
##     sosang_D_2 + sosang_F_2 + sosang_S_2 + cctv_count_1 + public_parking_2,
##     family = "binomial", data = df_var)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6390  -0.7288  -0.5404  -0.0001   2.4868
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)        0.3408     0.2697   1.263 0.206411
## gungu남구         -1.0895     0.2970  -3.669 0.000244 ***
## gungu동래구       -1.0953     0.2977  -3.680 0.000234 ***
## gungu사상구       -1.1725     0.2998  -3.912 9.17e-05 ***
## gungu서구         -1.4545     0.3382  -4.301 1.70e-05 ***
## gungu연제구      -17.7261   400.5773  -0.044 0.964704
## gungu영도구       -1.9834     0.2854  -6.950 3.65e-12 ***
## gungu중구         17.5823   320.0294   0.055 0.956186
## gungu해운대구     -1.2473     0.3057  -4.080 4.51e-05 ***
## building_13_1      2.1780     0.8784   2.479 0.013161 *
## building_10_1      1.6654     0.6848   2.432 0.015025 *
## building_16_1     -2.4005     0.5989  -4.008 6.12e-05 ***
## building_04_2     -2.5127     1.0748  -2.338 0.019397 *
## building_06_2     -1.4136     0.4737  -2.984 0.002843 **
## building_12_2      1.9091     0.8081   2.363 0.018149 *
## sosang_Q_1         4.1432     1.3733   3.017 0.002553 **
## sosang_D_2        -2.7017     1.7800  -1.518 0.129069
## sosang_F_2         1.7797     0.6617   2.689 0.007157 **
## sosang_S_2        -0.6705     0.5430  -1.235 0.216902
## cctv_count_1       0.5972     0.2656   2.248 0.024556 *
## public_parking_2  -3.0967     1.4589  -2.123 0.033782 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3176.6  on 2824  degrees of freedom
## Residual deviance: 2516.5  on 2804  degrees of freedom
## AIC: 2558.5
##
## Number of Fisher Scoring iterations: 16
```

회귀식의 각 계수는 오즈비로 한 단위 증가할 수록 exp(해당 계수)만큼 민원 다발 지역이 될 확률이 증가

모델 평가

```
lst_f1 = c(); lst_accuracy = c(); lst_recall = c(); lst_specificity = c(); lst_precision = c(); lst_auc <- c
()
sensitivities <- c()
specificities <- c()
group <- c()

for(i in 1:1000){
  set.seed(i)
  sample_num = sample(1:nrow(df_var), size = round(0.3 * nrow(df_var)))
  train <- df_var[-sample_num,]
  test <- df_var[sample_num,]
  model_glm <- glm(formula = y ~ gungu + building_13_1 + building_10_1 + building_16_1 +
    building_04_2 + building_06_2 + building_12_2 +
    sosang_Q_1 + sosang_D_2 + sosang_F_2 + sosang_S_2 + cctv_count_1 +
    public_parking_2,family='binomial', data=train)
  predicted <- predict(model_glm, newdata=test, type="response")

  roc_result <- roc(test$y , as.numeric(predicted))
  sensitivities <- c(sensitivities, roc_result$sensitivities)
  specificities <- c(specificities, roc_result$specificities)
  group <- c(group, rep(i,length(roc_result$specificities)))
  thred_which <- which((roc_result$sensitivities + roc_result$specificities)  == (max((roc_result$sensitivit
ies + roc_result$specificities))))
  confusion_mat <- table(test$y , ifelse(as.numeric(predicted)>=roc_result$thresholds[thred_which],1,0))
  accur <- (confusion_mat[1]+confusion_mat[4])/sum(confusion_mat)
  spec <- confusion_mat[1]/(confusion_mat[3]+confusion_mat[1])
  prec <- confusion_mat[4]/(confusion_mat[3]+confusion_mat[4])
  recal <- confusion_mat[4]/(confusion_mat[2]+confusion_mat[4])
  f1 <- (2*(prec*recal))/(prec+recal)

  lst_f1 = c(lst_f1, f1); lst_accuracy = c(lst_accuracy,accur);
  lst_recall = c(lst_recall,recal); lst_specificity = c(lst_specificity,spec); lst_precision = c(lst_precisi
on,prec);
  lst_auc <- c(lst_auc,roc_result$auc[1])
}
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
roc_df <- data.frame(lst_f1, lst_accuracy, lst_recall, lst_specificity, lst_precision,lst_auc)
```

시드를 바꿔가면 1000번의 검증을 시도함. 나온 결과값(f1 스코어, 정확도, 리콜, 특이도, 재현율, AUC)을 roc_df라는 테이블에 저장.

```
lapply(roc_df,mean)
```

```
## $lst_f1
## [1] 0.520238
##
## $lst_accuracy
## [1] 0.7205672
##
## $lst_recall
## [1] 0.6072491
##
## $lst_specificity
## [1] 0.7583484
##
## $lst_precision
## [1] 0.4654817
##
## $lst_auc
## [1] 0.7407963
```

```
lapply(roc_df,var)
```

```
## $lst_f1
## [1] 0.0004734282
##
## $lst_accuracy
## [1] 0.001328334
##
## $lst_recall
## [1] 0.005630657
##
## $lst_specificity
## [1] 0.004952517
##
## $lst_precision
## [1] 0.003149956
##
## $lst_auc
## [1] 0.0002877465
```

낮은 분산을 근거로 해당 모델은 과적합이 없는 안정성 있는 모델로 파악이 가능함.