기상청

[STEP4 분석모형 구축]

기상 데이터의 다중공선성 해결 방법

기상청

기상기후
빅데이터 분석 플랫폼

[분석교육] 다중공선성 문제

1. 다중공선성 문제 해결 함수

[분석 교육] 다중공선성 문제(1/2)

다중공선성 문제(Multicollinearity)는 독립변수들 간에 강한 상관관계가 나타나는 문제이며, 각 독립변수가 종속변수에 미치는 영향력을 왜곡 시킬 수 있기 때문에 분산팽창요인(VIF) 등을 통해 이를 확인 한 후 해결하여야 한다.

● 다중공선성 개념

- 다중공선성 문제(Multicollinearity)는 독립변수들 간에 강한 상관관계가 나타나는 문제이다.

	평균기온	최고기온	최저기온	일강수량	평균풍속	최대풍속	평균습도	최저습도	일조시간	일사량
평균기온	1.00	0.98	0.99	0.16	-0.06	-0.09	0.34	0.34	0.01	0.40
최고기온	0.98	1.00	0.94	0.12	-0.12	-0.13	0.28	0.24	0.11	0.47
최저기온	0.99	0.94	1.00	0.19	-0.01	-0.06	0.39	0.43	-0.06	0.32
일강수량	0.16	0.12	0.19	1.00	0.09	0.12	0.31	0.32	-0.33	-0.26
평균풍속	-0.06	-0.12	-0.01	0.09	1.00	0.90	-0.13	0.04	-0.05	-0.07
최대풍속	-0.09	-0.13	-0.06	0.12	0.90	1.00	-0.12	0.01	-0.02	-0.05
평균습도	0.34	0.28	0.39	0.31	-0.13	-0.12	1.00	0.88	-0.53	-0.37
최저습도	0.34	0.24	0.43	0.32	0.04	0.01	0.88	1.00	-0.59	-0.43
일조시간	0.01	0.11	-0.06	-0.33	-0.05	-0.02	-0.53	-0.59	1.00	0.80
일사량	0.40	0.47	0.32	-0.26	-0.07	-0.05	-0.37	-0.43	0.80	1.00

<그림 1>기상 데이터 독립변수 간 상관계수

- 예를 들어, 기상 데이터 간의 상관계수는 <그림1>과 같다.
- 평균기온-최고기온, 평균기온-최저기온, 최고기온-최저기온 간은 0.9 이상인 강한 상관관계가 나타난다.
- 다중공선성을 가지는 독립변수들을 회귀 분석에 활용할 경우, 각 독립변수 상호간에는 상관관계가 없어야 한다는 회귀분석의 가정을 위반하게 되고, 회귀계수가 데이터를 제대로 설명하지 못하는 문제가 발생할 수 있다.
- 또한, 독립변수들간의 상관성때문에 해당 각 독립변수가 종속변수에 미치는 영향력을 정확하게 설명하지 못하는 문제가 발생할 수 있다.
- 그러므로 분석 시에는 독립변수간의 다중공선성을 제거하여 종속변수에 영향을 미치는 가장 중요한 변수들만을 활용하여야 한다.

[분석 교육] 다중공선성 문제(2/2)

다중공선성 확인 방법

- 결정계수 R^2값은 높지만 독립변수의 p-value 값이 커서 개별 인자들이 유의하지 않는 경우
- 독립변수들간의 상관계수를 구하여 상관성이 높은 경우
- 분산팽창요인(Variance Inflation Factor)을 구하여 값이 10이 넘는 경우
  - 분산팽창요인(Variance Inflation Factor)은 다중공선성을 판단하기 위해 사용되는 계수로써 VIF 수식의 값이 10 이상이면 다중공선성이 존재하는 것으로 판단한다.
  - 변수 X를 종속변수로 지정하고 나머지 변수를 독립변수로 하여 회귀분석을 수행한 뒤, 결과로 도출된 모형의 결정계수 (R^2)를 활용하여 다음 계산식으로 VIF를 구한다.

$$\mathrm{VIF_i} = \frac{1}{1 - R_i^2}$$

다중공선성 문제 해결 방법

- 상관관계가 높은 독립 변수 중 하나 혹은 일부를 제거한다.
- 변수를 변형시키거나 새로운 관측치를 이용한다.
- 자료를 수집하는 현장의 상황을 보아 상관관계의 이유를 파악하여 해결한다.

참고 자료 : 위키백과

1. 다중공선성 문제 해결 함수(1/4)

- **cor**

  - 변수들간의 상관계수를 산출

▪ **Usage**

cor(x, method = “Pearson”, …)

- x : 숫자 벡터, 행렬, 데이터 프레임
- method : 상관계수의 종류를 지정
  : 피어슨(Pearson), 켄달(Kendall), 스피어만(Spearman)
  : 기본값은 피어슨(Pearson)

▪ **Examples**

```
Cor <- cor(DATA[,!(names(DATA) %in% c("STN_ID","TM","SUM_SML_EV"))])
```

```
> Cor <- cor(DATA[,!(names(DATA) %in% c("STN_ID","TM","SUM_SML_EV"))])
> head(Cor)
           TA_AVG   TA_MAX    TA_MIN   SUM_RN    WS_AVG    WS_MAX   HM_AVG
TA_AVG  1.000000  0.98169  0.985247 0.159266 -0.060841 -0.090234  0.34190
TA_MAX  0.981688  1.00000  0.940713 0.122972 -0.119422 -0.134929  0.27949
TA_MIN  0.985247  0.94071  1.000000 0.186498 -0.014390 -0.055623  0.39453
SUM_RN  0.159266  0.12297  0.186498 1.000000  0.087562  0.116370  0.31376
WS_AVG -0.060841 -0.11942 -0.014390 0.087562  1.000000  0.899057 -0.13135
WS_MAX -0.090234 -0.13493 -0.055623 0.116370  0.899057  1.000000 -0.11784
           HM_MIN    SUM_SS    SUM_SI    TD_AVG    PV_AVG    PA_AVG    PS_AVG
TA_AVG 0.344547  0.013691  0.397379  0.876752  0.845585 -0.501093 -0.704926
TA_MAX 0.235833  0.105621  0.470593  0.841537  0.804079 -0.486507 -0.681197
TA_MIN 0.427476 -0.064787  0.320779  0.882880  0.858225 -0.497098 -0.699893
SUM_RN 0.323218 -0.325889 -0.261197  0.211341  0.234725 -0.178609 -0.225103
WS_AVG 0.043338 -0.050909 -0.065884 -0.086178 -0.078620 -0.084678 -0.061409
WS_MAX 0.013757 -0.023842 -0.051583 -0.106403 -0.099353 -0.097994 -0.057839
           PS_MAX    PS_MIN SD_HR3_MAX SD_TOT_MAX CA_TOT_AVG CA_MID_AVG
TA_AVG -0.737055 -0.653437  -0.172937  -0.188669   0.248887   0.183079
TA_MAX -0.710982 -0.634164  -0.187281  -0.205621   0.161436   0.094607
TA_MIN -0.735438 -0.645238  -0.158721  -0.169941   0.317364   0.253219
SUM_RN -0.195107 -0.246896   0.053428   0.018216   0.328643   0.317658
WS_AVG -0.027725 -0.093761   0.043452   0.032647   0.056505   0.100148
WS_MAX -0.017520 -0.099795   0.053272   0.037146   0.022130   0.065077
           TS_AVG    TS_MAX    TS_MIN         HT     RAIN     WS_2m
TA_AVG  0.907656  0.822777  0.905334 -0.0462513 0.105754 -0.064052
TA_MAX  0.900187  0.848129  0.873388 -0.0401111 0.042303 -0.122951
TA_MIN  0.886899  0.778590  0.907763 -0.0468523 0.154697 -0.017373
SUM_RN  0.114283  0.014705  0.184396  0.0069611 0.410493  0.087135
WS_AVG -0.081646 -0.104280 -0.063480  0.0629702 0.106206  0.996215
WS_MAX -0.104218 -0.118197 -0.092961  0.0844748 0.115127  0.891257
```

1. 다중공선성 문제 해결 함수(2/4)

- **abs**

  - 절대값을 반환

  ▪ **Usage**
  - x : 숫자 벡터 또는 배열

  ▪ **Examples**
  Abs_Cor <- abs(as.data.frame(Cor))

```
> Abs_Cor <- abs(as.data.frame(Cor))
> head(Abs_Cor)
          TA_AVG   TA_MAX   TA_MIN   SUM_RN   WS_AVG   WS_MAX  HM_AVG   HM_MIN
TA_AVG 1.000000 0.98169 0.985247 0.159266 0.060841 0.090234 0.34190 0.344547
TA_MAX 0.981688 1.00000 0.940713 0.122972 0.119422 0.134929 0.27949 0.235833
TA_MIN 0.985247 0.94071 1.000000 0.186498 0.014390 0.055623 0.39453 0.427476
SUM_RN 0.159266 0.12297 0.186498 1.000000 0.087562 0.116370 0.31376 0.323218
WS_AVG 0.060841 0.11942 0.014390 0.087562 1.000000 0.899057 0.13135 0.043338
WS_MAX 0.090234 0.13493 0.055623 0.116370 0.899057 1.000000 0.11784 0.013757
          SUM_SS   SUM_SI   TD_AVG   PV_AVG   PA_AVG   PS_AVG   PS_MAX   PS_MIN
TA_AVG 0.013691 0.397379 0.876752 0.845585 0.501093 0.704926 0.737055 0.653437
TA_MAX 0.105621 0.470593 0.841537 0.804079 0.486507 0.681197 0.710982 0.634164
TA_MIN 0.064787 0.320779 0.882880 0.858225 0.497098 0.699893 0.735438 0.645238
SUM_RN 0.325889 0.261197 0.211341 0.234725 0.178609 0.225103 0.195107 0.246896
WS_AVG 0.050909 0.065884 0.086178 0.078620 0.084678 0.061409 0.027725 0.093761
WS_MAX 0.023842 0.051583 0.106403 0.099353 0.097994 0.057839 0.017520 0.099795
       SD_HR3_MAX SD_TOT_MAX CA_TOT_AVG CA_MID_AVG   TS_AVG   TS_MAX   TS_MIN
TA_AVG   0.172937   0.188669   0.248887   0.183079 0.907656 0.822777 0.905334
TA_MAX   0.187281   0.205621   0.161436   0.094607 0.900187 0.848129 0.873388
TA_MIN   0.158721   0.169941   0.317364   0.253219 0.886899 0.778590 0.907763
SUM_RN   0.053428   0.018216   0.328643   0.317658 0.114283 0.014705 0.184396
WS_AVG   0.043452   0.032647   0.056505   0.100148 0.081646 0.104280 0.063480
WS_MAX   0.053272   0.037146   0.022130   0.065077 0.104218 0.118197 0.092961
             HT      RAIN    WS_2m
TA_AVG 0.0462513 0.105754 0.064052
TA_MAX 0.0401111 0.042303 0.122951
TA_MIN 0.0468523 0.154697 0.017373
SUM_RN 0.0069611 0.410493 0.087135
WS_AVG 0.0629702 0.106206 0.996215
WS_MAX 0.0844748 0.115127 0.891257
```

1. 다중공선성 문제 해결 함수(3/4)

- **h2o::h2o.varimp**

  - 모형의 변수 중요도를 산출

  ▪ **Usage**

  h2o.varimp(object)

  - object : H2O 모델 객체

  ▪ **Examples**

  h2o.varimp(fit)

```
> varimp <-h2o.varimp(fit)
> head(varimp)
Variable Importances:
    variable relative_importance scaled_importance percentage
1     TS_AVG      1006930.625000          1.000000   0.134883
2     TS_MAX      1002898.875000          0.995996   0.134343
3     SUM_SI       575601.062500          0.571639   0.077104
4     PS_MAX       549429.062500          0.545647   0.073599
5 CA_TOT_AVG       539076.937500          0.535367   0.072212
6     SUM_SS       474950.125000          0.471681   0.063622
```

- **order**

  - 데이터를 정렬하기 위한 순서 반환

  ▪ **Usage**

  - x : 정렬할 데이터
  - decreasing : 내림차순 여부, 기본값은 FALSE

  ▪ **Examples**

  tmp <- tmp[order(-tmp$Pearson),]

```
> tmp <- subset(Abs_Cor, variable != "y" & variable != finalvar$variable[j], select = c("var
> colnames(tmp) <- c("variable", "Pearson")
> head(tmp)
  variable  Pearson
1   TA_AVG 0.907656
2   TA_MAX 0.900187
3   TA_MIN 0.886899
4   SUM_RN 0.114283
5   WS_AVG 0.081646
6   WS_MAX 0.104218
> tmp <- tmp[order(-tmp$Pearson),]
> head(tmp)
   variable Pearson
23   TS_MIN 0.96600
22   TS_MAX 0.93896
11   TD_AVG 0.91709
1    TA_AVG 0.90766
2    TA_MAX 0.90019
3    TA_MIN 0.88690
```

1. 다중공선성 문제 해결 함수(4/4)

- **while 반복문**

  - 특정 조건이 TRUE일때 블록안의 문장을 반복 수행

▪ **Usage**

```
while (조건) {
  조건이 TRUE일 때 수행할 문장
 }
```

▪ **Examples**

```
finalvar <- varimp
j=1 # 초기값 셋팅
while(j < length(finalvar$variable)) {
      print(paste0(j, " calculating..."))
      tmp <- subset(Abs_Cor, variable != "y" & variable != finalvar$variable[j],
                    select = c("variable", finalvar$variable[j]))
       colnames(tmp) <- c("variable", "Pearson")
       tmp <- tmp[order(-tmp$Pearson),]
       tmp <- subset(tmp, tmp$Pearson > 0.45 )
       finalvar <- merge(finalvar, tmp, by="variable", all.x=TRUE, all.y=FALSE)
       finalvar <- subset(finalvar, is.na(Pearson), select=-Pearson)
       finalvar <- as.data.frame(finalvar[order(-finalvar$scaled_importance),])
       j = j + 1
}
```

```
> finalvar <- varimp
> j=1
> while(j < length(finalvar$variable)) {
+   print(paste0(j, " calculating..."))
+   tmp <- subset(Abs_Cor, variable != "y" & variable != finalvar$variable[j], s
+   colnames(tmp) <- c("variable", "Pearson")
+   tmp <- tmp[order(-tmp$Pearson),]
+   tmp <- subset(tmp, tmp$Pearson > 0.45 )
+   finalvar <- merge(finalvar, tmp, by="variable", all.x=TRUE, all.y=FALSE)
+   finalvar <- subset(finalvar, is.na(Pearson), select=-Pearson)
+   finalvar <- as.data.frame(finalvar[order(-finalvar$scaled_importance),])
+   j = j + 1
+ }
[1] "1 calculating..."
[1] "2 calculating..."
[1] "3 calculating..."
[1] "4 calculating..."
[1] "5 calculating..."
[1] "6 calculating..."
```

본 문서의 내용은 기상청의 날씨마루(http://big.kma.go.kr) 내
R 프로그래밍 교육 자료입니다.