[융합]데이터마이닝&정보디자인
과제 #02
선형회귀분석 & 로지스틱 회귀분석

2021.04.28

2017204081 최수지

목차

1. 일상생활에서 볼 수 있는 Odd 사례를 한 가지 찾아보고 설명하시오.

Odd : 성공 범주에 속할 확률(positive class)을 'p' 로 칭할 때,
다음과 같은 식을 얻을 수 있다.

$$\text{Odds} = \frac{p}{1-p}$$

Ex) 2020-2021 프리미어리그에서 총 20개의 팀이 트로피를 차지하기 위하여 경기를 펼치고 있다. 맨체스터 유나이티드는 현재 프리미어리그에서 2위를 유지하고 있으며, 프리미어리그에서 현재 3위를 유지하고 있는 레스터 시티 FC를 상대로 2021년 5월 12일에 경기를 진행할 예정이다.
맨체스터 유나이티드는 레스터 시티 FC를 상대로 프리미어리그에서 현재까지 총 13번의 경기를 치뤘고, 8번의 승리와 1번의 패배, 그리고 4번의 무승부를 하였다.
경기를 치루는 구성원의 수와 경기 승리의 규칙 및 팀 내 구성원들의 변경은 고려하지 않는다. 또한 무승부는 승리/패배 중 어느 것으로도 인정하지 않을 때, 레스터 시티 FC에 대한 맨체스터 유나이티드가 승리할 Odd를 구해보자.

$$p = 0.6154\text{(소수 다섯 번째 자리에서 반올림)},$$
$$\text{Odds} = \frac{0.6154}{1-0.6154} = \frac{0.6154}{0.3846} = 1.6001\text{(소수 다섯 번째 자리에서 반올림)}$$

이를 통해 맨체스터 유나이티드와 레스터 시티 FC가 경기를 진행할 경우를 생각해보자. 조심해야 할 점은, 맨체스터 유나이티드가 승리하지 않는다고 해서 레스터 시티 FC가 승리한다는 것은 아니다(무승부 또한 분모에 들어가므로). 그러므로 맨체스터 유나이티드에 대한 레스터 시티 FC가 승리할 Odd를 구하여 둘을 비교 해 본다.

$$p = 0.0769\text{(소수 다섯 번째 자리에서 반올림)},$$
$$\text{Odds} = \frac{0.0769}{1-0.0769} = \frac{0.0769}{0.9231} = 0.0833\text{(소수 다섯 번째 자리에서 반올림)}$$

프리미어리그에서 레스터 시티 FC에 대한 맨체스터 유나이티드의 승리 Odd가 약 1.6이며 맨체스터 유나이티드에 대한 레스터 시티 FC의 승리 Odd가 약 0.08임을 알 수 있다.

2. 다중회귀분석과 로지스틱 회귀분석 실습

2.1 (선형회귀분석) 적합한 데이터를 찾으시오.

다중선형회귀분석에 사용할 DataSet

Student Grade Prediction

다음은 DataSet과 관련된 설명이다.

"학생의 여러 속성값을 통한 수학 과목 성취도 예측"

독립 변수 개수(30개), 종속 변수 개수(각 학년 성적, 3개)

1. school (GP:Gabriel Pereira / MS:Mousinho da Silveira)
2. sex (F:female / M:male)
3. age (15~22)
4. address (U:도시, R:시골)
5. famsize (LE3:가족 수가 3이하 / GT3:가족 수가 3초과)
6. Pstatus (T:아빠엄마가 같이 삶/A:아빠 엄마가 따로 삶)
7. Medu (엄마 학력)
8. Fedu (아빠 학력)
9. Mjob (엄마 직업)
10. Fjob (아빠 직업)
11. reason (이 학교를 택한 이유)
12. guardian (학생의 보호자)
13. traveltime (등교하는데 걸리는 시간)
14. studytime (일주일에 몇시간 공부하는지)
15. failures (낙제 횟수)
16. schoolsup (추가적인 교육 지원)
17. famsup (가족의 교육 지원)
18. paid (수학 과목 유료 수업 추가 신청)
19. activities (교외 활동)
20. nursery (보육원 참여 여부)
21. higher (고등 교육 학습 의사)
22. internet (집 인터넷 여부)
23. romantic (이성친구 여부)
24. famrel (가족 관계 점수)
25. freetime (수업 이후 자유 시간)
26. goout (친구들과의 외출)
27. Dalc (workday기준 알코올 섭취 여부)
28. Walc (weekend기준 알코올 섭취 여부)
29. health (건강 상태 점수)
30. absence (결석 횟수)

2.2 (선형회귀) 탐색적 데이터 분석(EDA)를 진행하고 구축한 모델에 대하여 평가하시오.

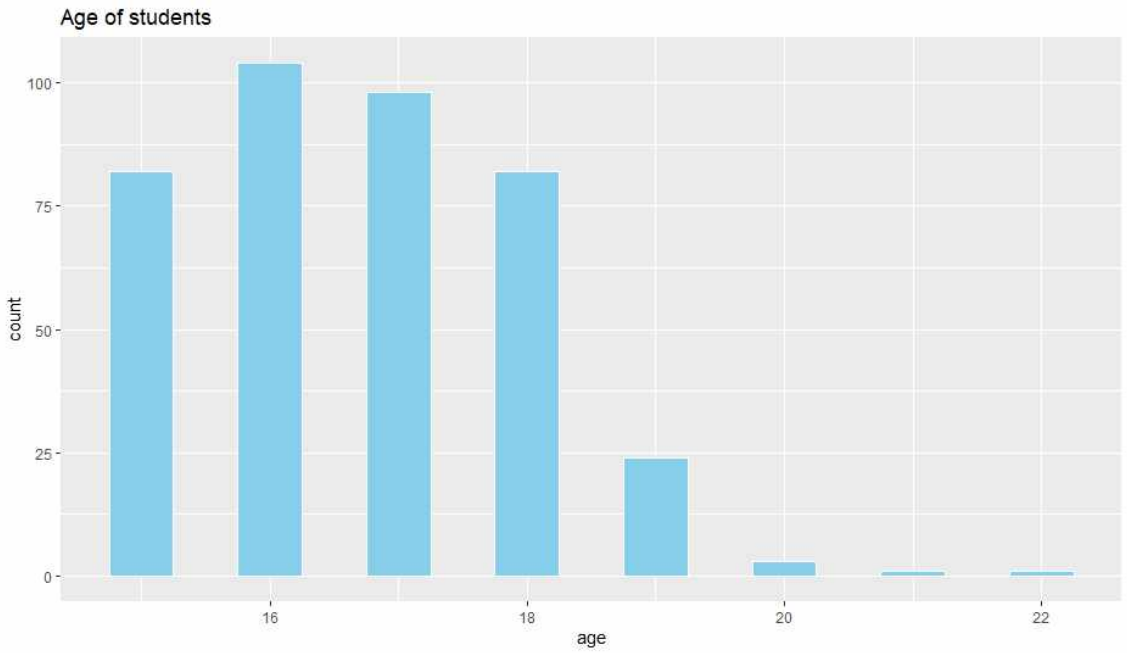
```
> sum(is.na(csv1))
[1] 0
#결측치 존재X
> dim(csv1)
[1] 395  33
#행 개수 395개, 열 개수 33개
```

```
#이진 값(YES:1/NO:0으로 변환)
csv1$schoolsup <- ifelse(csv1$schoolsup == 'yes', 1, 0)
csv1$famsup <- ifelse(csv1$famsup == 'yes', 1, 0)
csv1$paid <- ifelse(csv1$paid == 'yes', 1, 0)
csv1$activities <- ifelse(csv1$activities == 'yes', 1, 0)
csv1$nursery <- ifelse(csv1$nursery == 'yes', 1, 0)
csv1$higher <- ifelse(csv1$higher == 'yes', 1, 0)
csv1$internet <- ifelse(csv1$internet == 'yes', 1, 0)
csv1$romantic <- ifelse(csv1$romantic == 'yes', 1, 0)
```

```
#범주형 변수(이진값처럼 변환)
student2 <- data.frame(csv1)
student2$school <- ifelse(student2$school == 'GP', 1, 0) #학교가 GP면 1
student2$sex <- ifelse(student2$sex == 'F', 1, 0) #여성이면 1
student2$address <- ifelse(student2$address == 'U', 1, 0) #도시 거주면 1
student2$famsize <- ifelse(student2$famsize == 'GT3', 1, 0) #가족 수가 3명보다 많으면 1
student2$Pstatus <- ifelse(student2$Pstatus == 'T', 1, 0) #부모가 같이 살면 1
student2$guardian <- ifelse(student2$guardian == 'mother', 1, 0) #보호자가 엄마면 1
head(student2)
```

YES/NO값을 가진 독립변수는 YES를 1로, NO를 0으로 변환하였습니다. 이를 csv1라 하겠습니다. 여기서 카테고리형 독립변수임에도 binary한 값을 띄는 경우에 각 변수의 기준을 잡아 1과 0으로 변환하였습니다. 이를 student2라 하겠습니다.
csv1를 더미변수로 변환하여 student1을 만들었으나, 모델링 과정 도중 적합하지 않다는 결과값이 나와 다음으로 넘어갑니다. EDA는 csv1기준으로, 모델링은 student2 기준으로 진행하고자 합니다.

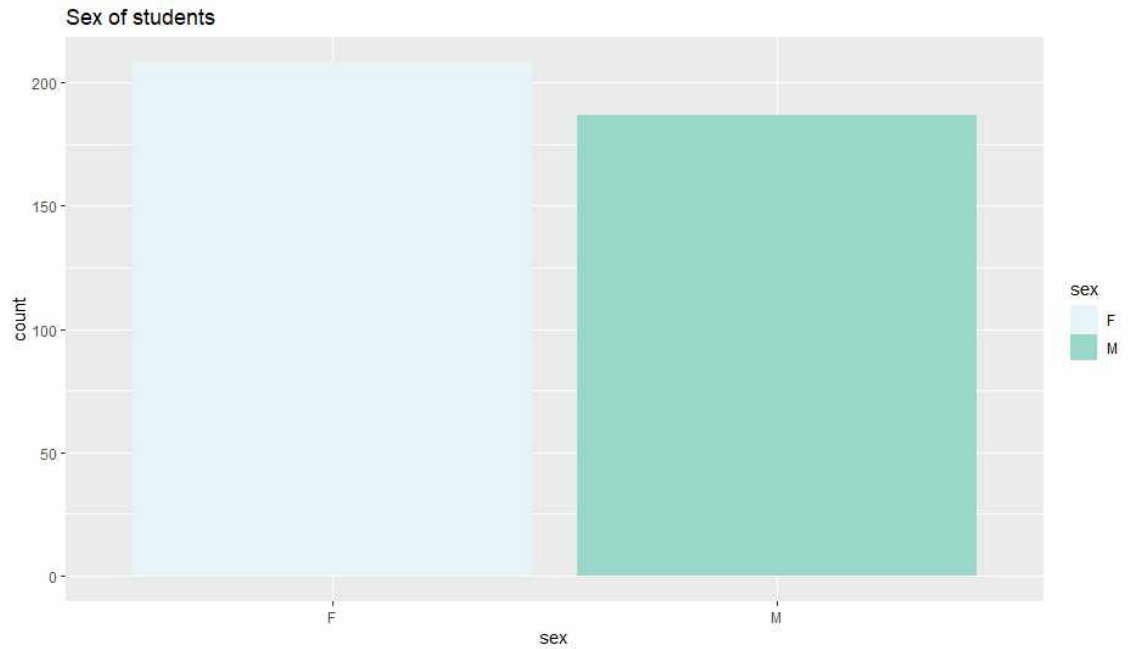
```
table(csv1$age)
var(csv1$age)
ggplot(data=csv1,aes(x=age))+
  geom_histogram(binwidth = 0.50, fill='skyblue', color='white')+
  ggtitle("Age of students")
```




```
table(csv1$age)
var(csv1$age)
ggplot(data=csv1,aes(x=age))+
  geom_histogram(binwidth = 0.50, fill='skyblue', color='white')+
  ggtitle("Age of students")
```

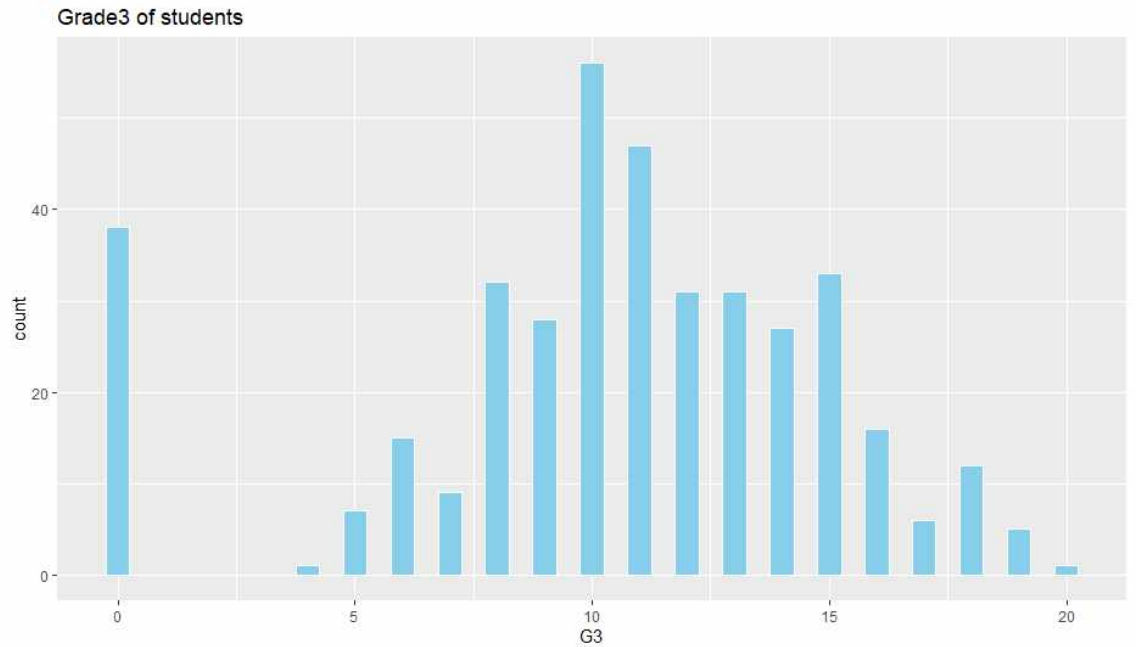

학생들의 나이 분포를 조사하였습니다. 이 차트는 학생들의 나이가 주로 15살~18살 사이에 값이 집중되어 있음을 쉽게 알아 볼 수 있습니다.

```
table(csv1$sex)
ggplot(data=csv1,aes(x=sex,fill=sex))+
  geom_bar()+
  scale_fill_brewer(palette=2)+
  ggtitle("Sex of students")
```



학생들의 성별을 조사하였습니다. 이 차트를 통하여 전체 조사 대상 중 여학생들의 수가 남학생들에 비해 조금 더 많음을 알 수 있습니다.

```
table(csv1$G3)
ggplot(data=csv1,aes(x=G3))+
  geom_histogram(binwidth = 0.50, fill='skyblue', color='white')+
  ggtitle("Grade3 of students")
```



추가적으로 이 차트에선, G3값이 0인 학생들이 꽤나 존재함을 바로 알아볼 수 있습니다. 가장 많이 받은 Grade값은 10임을 바로 알아볼 수 있습니다.

모든 예측 변수를 적용한 모델에서 변수선택법을 따로 적용하지 않고, G1/G2를 제외한 (지난 성적은 이후 성적에 큰 영향을 미칠것이므로, 그 이외의 요인 탐색을 위하여)P-value가 유의수준(0.1)보다 낮은 예측 변수 8개를 임의로 선택하여 구축한 모델을 살펴보자.

```
Call:
lm(formula = G3 ~ sex + Medu + studytime + failures + schoolsup +
    romantic + goout + absences, data = student2)

Residuals:
     Min       1Q   Median       3Q      Max
-12.8826  -1.9777   0.4914   2.8315   8.1945

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 10.91119    0.98803  11.043  < 2e-16 ***
sex         -1.12077    0.45011  -2.490  0.01320 *
Medu         0.55404    0.20041   2.765  0.00597 **
studytime    0.44443    0.26710   1.664  0.09694 .
failures    -1.85069    0.29847  -6.201 1.45e-09 ***
schoolsup   -1.08844    0.63198  -1.722  0.08582 .
romantic    -1.13138    0.45499  -2.487  0.01332 *
goout       -0.46704    0.19066  -2.450  0.01474 *
absences     0.04472    0.02681   1.668  0.09615 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.147 on 386 degrees of freedom
Multiple R-squared:  0.1971,    Adjusted R-squared:  0.1805
F-statistic: 11.85 on 8 and 386 DF,  p-value: 3.975e-15
```

[선형회귀 모델 평가 기준]

Pr(>|t|)_p-value

sex(0.01320), Medu(0.00597), failures(1.45e-09), romantic(0.01332), goout(0.01474)는 유의수준 0.05보다 훨씬 작은 값으로 구성되어 있다. 따라서 예측 변수들이 영향이 없을 것이라는 귀무가설을 기각하고 영향이 있다는 대립가설을 채택하게 된다.

R-squared_결정계수

0.1971으로 약 20%정도의 설명력을 가진다. 아주 적은 수치이다.

R-squared 결정계수 값이 의미가 없을 정도로 나왔다. 따라서 G1/G2를 복원하여 모델링을 다시 진행해본다.

모든 예측 변수를 적용한 모델에서 변수선택법을 따로 적용하지 않고, P-value가 유의수준(0.1)보다 낮은 예측 변수 6개를 임의로 선택하여 구축한 모델을 살펴보자.

```
Call:
lm(formula = G3 ~ age + activities + famrel + absences + G1 +
    G2, data = student2)

Residuals:
     Min       1Q   Median       3Q      Max
-8.8823 -0.4475  0.2760  1.0104  3.9410

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.28769    1.38534   0.208 0.835597
age         -0.21654    0.07695  -2.814 0.005139 **
activities  -0.35893    0.19015  -1.888 0.059816 .
famrel       0.36624    0.10598   3.456 0.000610 ***
absences     0.04384    0.01201   3.651 0.000297 ***
G1           0.16158    0.05489   2.944 0.003436 **
G2           0.97705    0.04879  20.025  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.875 on 388 degrees of freedom
Multiple R-squared:  0.8351,    Adjusted R-squared:  0.8326
F-statistic: 327.5 on 6 and 388 DF,  p-value: < 2.2e-16
```

[선형회귀 모델 평가 기준]

잔차(Residuals)_학습모델의 오차(e) : (아래 Normal Q-Q를 통하여 잔차 정도 파악)

계수(Coefficient)_선형회귀의 베타 값 : 각 변수들의 계수값 표에 기재

Pr(>|t|)_p-value

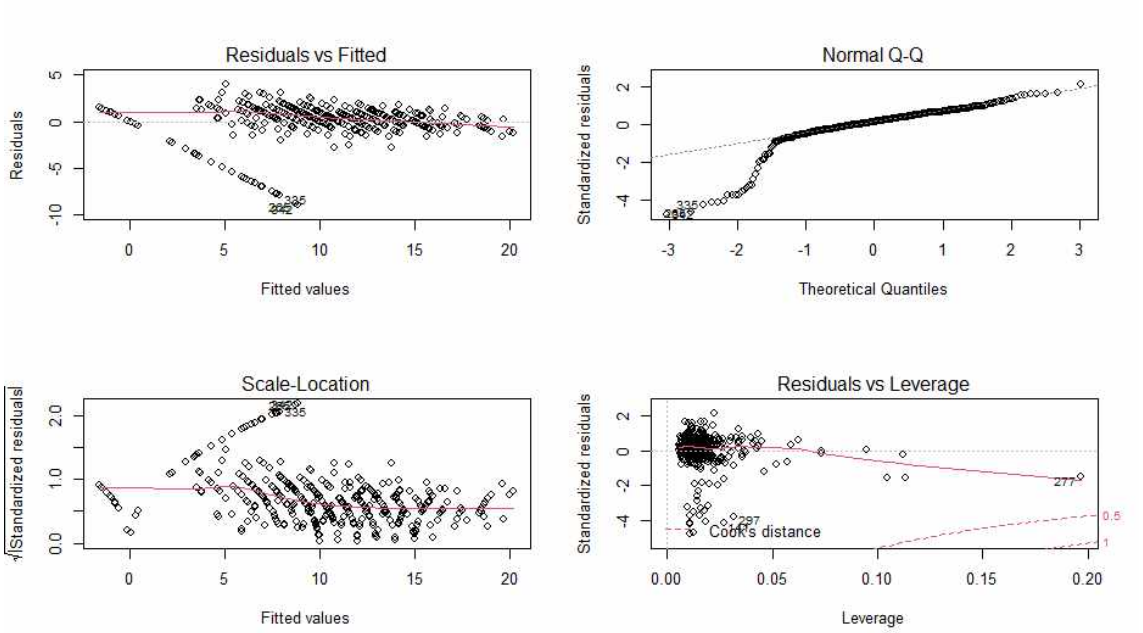
age(0.005139), famrel(0.000610), absences(0.000297), G1(0.000297), G2(0.003436)들은 유의수준 0.05보다 훨씬 작은 값으로 구성되어 있다. 따라서 예측 변수들이 영향이 없을 것이라는 귀무가설을 기각하고 영향이 있다는 대립가설을 채택하게 된다.

R-squared_결정계수

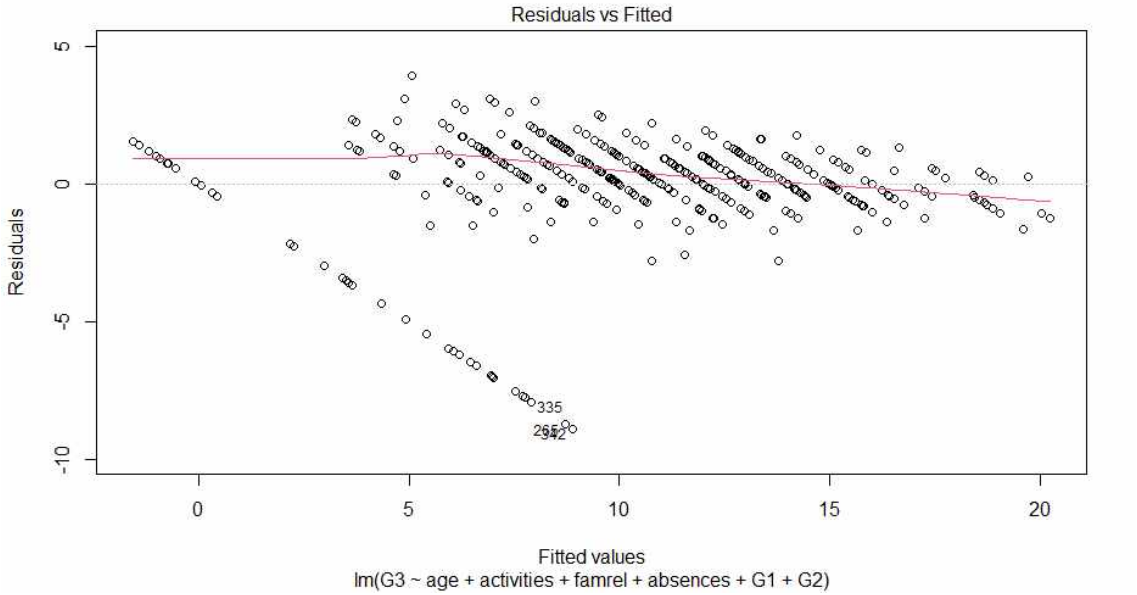
0.8351으로 약 84%정도의 설명력을 가진다.

```
par(mfrow = c(2,2))
plot(student2Model) # 모델에 대한 전체 plot
plot(csv1Model, which = 1) #Residuals vs Fitted
plot(csv1Model, which = 2) #Normal Q-Q
plot(csv1Model, which = 3) #Scale-Location
plot(csv1Model, which = 4) #Cook's distance
plot(csv1Model, which = 5) #Residuals vs Leverage
```

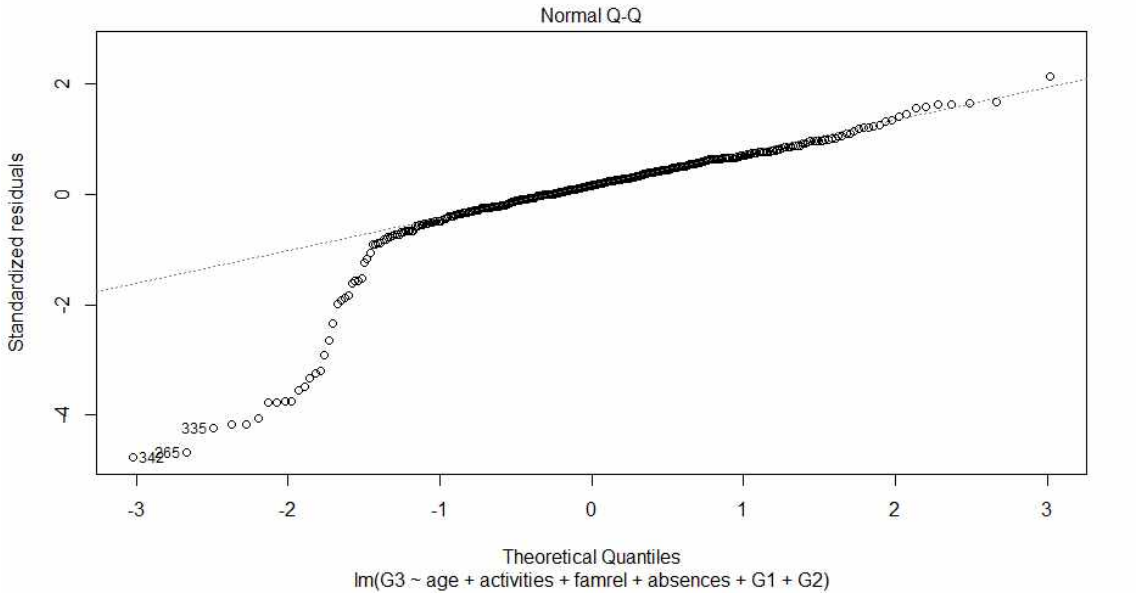
[전체 plot]



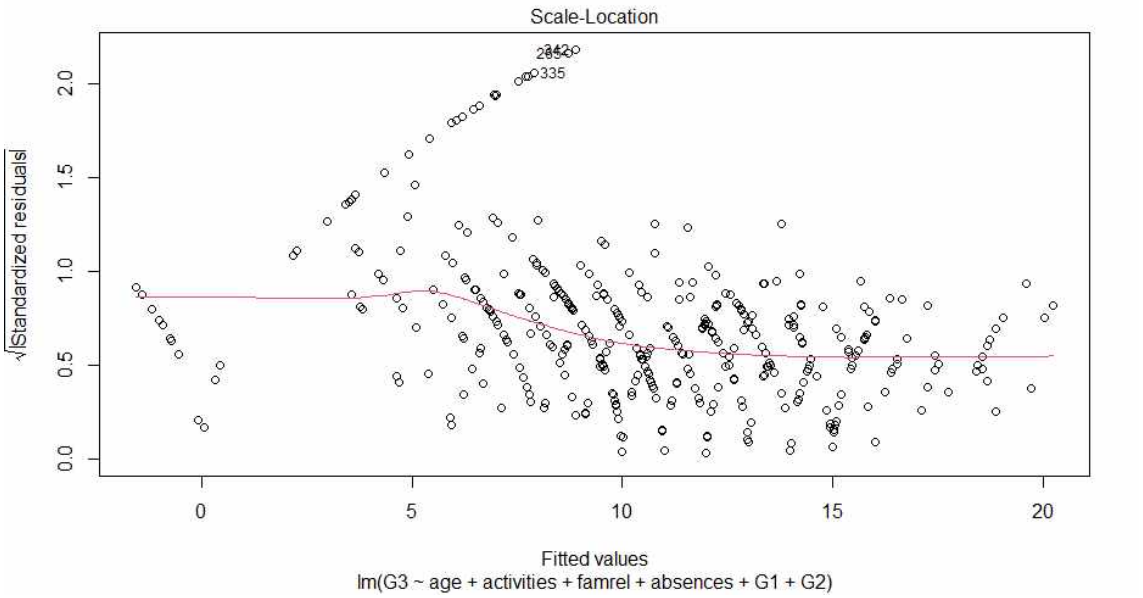
[실제와 예측값 사이의 잔차 확인]
여기서 잔차와 예측값 간의 특정 패턴이 두 개가 관찰된다. 따라서 선형성 가정을 완벽하게는 충족하지는 못한 것으로 판단된다.



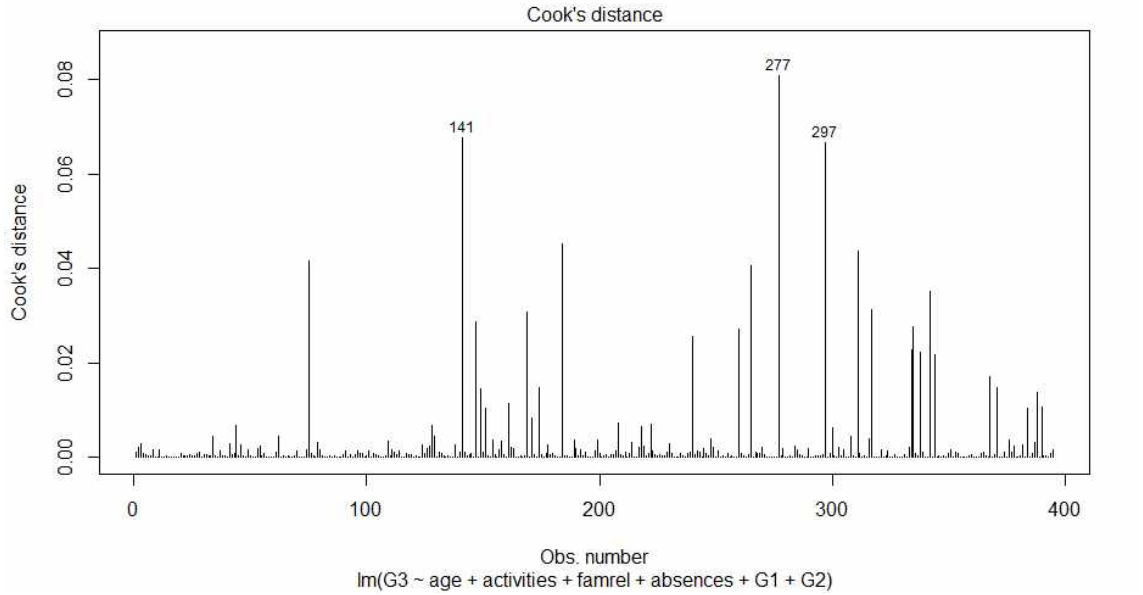
[정규분포에 대한 잔차 비교]
대부분의 점들이 대각선 부근에 분포하고 있으나, 대각선을 벗어난 점들의 수도 꽤 많다. 이를 통하여 정규성의 가정을 온전히 충족하고 있다고 보기는 어렵다.



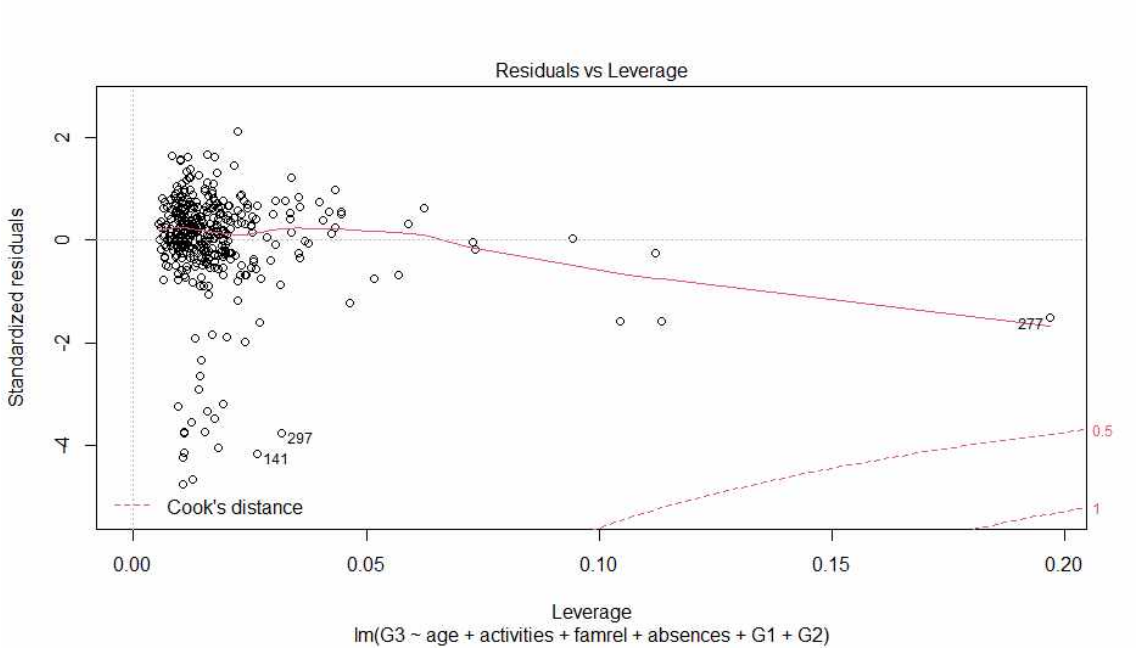
[선형모델의 예측값과 잔차 간의 관계 비교]
수평 추세선은 관찰되나, 일부 값들이 대각선으로 분포하는 것을 알 수 있다.
이를 통하여 등분산 가정을 온전히 충족하고 있다고 보기는 어렵다.



[개별 관측치가 선형모델의 파라미터(베타 제로, 베타 원)에 끼친 영향도 측정]



[독립변수의 극단적 치우침 확인 plot]
생성한 모델으로는 잘 설명되지 않는 관측치를 확인할 수 있다.



위와 같은 모델 평가를 거치고 선형회귀 모델이 적합한지를 다시 점검해보자.

유의수준(0.05)를 기준으로, age/famrel/absences/G1/G2 는 유의한 변수임을 확인하였다.
또한, R-squared 값이 0.8351로 선형모델이 83.51%의 설명력을 가지고 있다.
G1/G2를 제거하였을 때는
sex/Medu/studytime/failures/schoolsup/romantic/goout/absences가 비교적 유의한
변수였으나, absences를 제외하고 전혀 다른 결과를 가진다.

2.3 선형회귀 결과해석 : 예측결과로 MSE(Mean Squared Error)와 MAE(Mean absolute Error)를 측정하고 설명하시오.

```
student2Predict <- predict(student2Model, newdata = student2Test)
mean((student2Test$G3 - student2Predict)**2)

> mean((student2Test$G3 - student2Predict)**2)
[1] 3.514688 (MSE)
```

```
library(caret) # MAE를 보다 손쉽게 계산할 수 있도록 도와주는 라이브러리
caret::MAE(student2Predict,student2Test$G3)

> caret::MAE(student2Predict,student2Test$G3)
[1] 1.162323
```

MSE(Mean Squared Error) : 모든 예측값에서 실제 값을 뺀 값을 다 더한 뒤 이를 제곱하고 평균을 낸 값.
MSE는 F분포, 즉 추정된 회귀식의 설명하는 영역이 설명하지 못하는 영역에 비하여 얼마나 설명할 수 있는지 알 수 있는 값을 구하기 위하여 쓰인다. p-value는 유의 수준보다 낮아야 귀무가설을 기각하고 대립가설을 채택할 확률이 높지만, F 분포는 클수록 귀무가설을 기각할 가능성이 높아진다.
이 말을 쉽게 정리하자면, 데이터를 잘 설명하고 잘 예측하는 모델을 만들고 싶다면 MSE를 낮추는 방향으로 해야 한다.
현재 선형회귀 모델에서 MSE = 3.514688으로 측정하였다.

MAE(Mean absolute Error) : 절댓값을 씌운 오차를 모두 더하여 평균을 낸 값.
현재 선형회귀 모델에서 MAE = 1.162323으로 측정하였다.

2.1 (로지스틱 회귀분석) 적합한 데이터를 찾으시오.

로지스틱 회귀분석에 사용할 Data Set

“Mobile Price Classification”

다음은 Data Set과 관련된 설명이다.

“핸드폰의 속성 값을 통한 가격 범주 분류”

독립 변수 20개, 종속 변수 2개

1. battery_power (배터리 mAh)
2. blue (블루투스 기능 여부)
3. clock_speed (cpu 속도)
4. dual_sim (듀얼 SIM지원 여부)
5. fc (후면 카메라 화소)
6. four_g (4G 여부)
7. int_memory (내장메모리 기가바이트 단위)
8. m_dep (핸드폰 두께 센티미터 단위)
9. mobile_wt (핸드폰 무게)
10. n_cores (cpu 개수)
11. pc (전면카메라 화소)
12. px_height (해상도 높이)
13. px_width (해상도 너비)
14. ram (램 메가바이트 단위)
15. sc_h (스크린 높이)
16. sc_w (스크린 너비)
17. talk_time (배터리 사용가능 시간)
18. three_g (3G)
19. touch_screen (터치스크린 여부)
20. wifi (와이파이 여부)
21. price_range (0-3, 0:low 1:medium / 2:high 3:very high)

=> 이진 분류를 위한 변수 range 추가

22. range(0:low / 1:high)

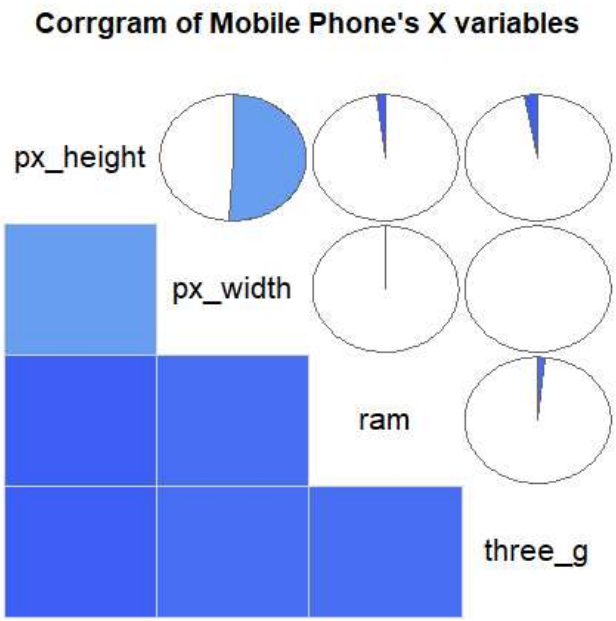
2.2 (로지스틱회귀) 탐색적 데이터 분석(EDA)를 진행하고 구축한 모델에 대하여 평가하시오.

```
> sum(is.na(csv2))
[1] 0
#결측치 존재X
> dim(csv2)
[1] 2000   21
#행 개수 2000개, 열 개수 21개
```

```
> table(price$range)

   0    1
1000 1000
#0,1에 정확하게 50:50으로 되어있으므로 스케일링이 필요 없다!
```

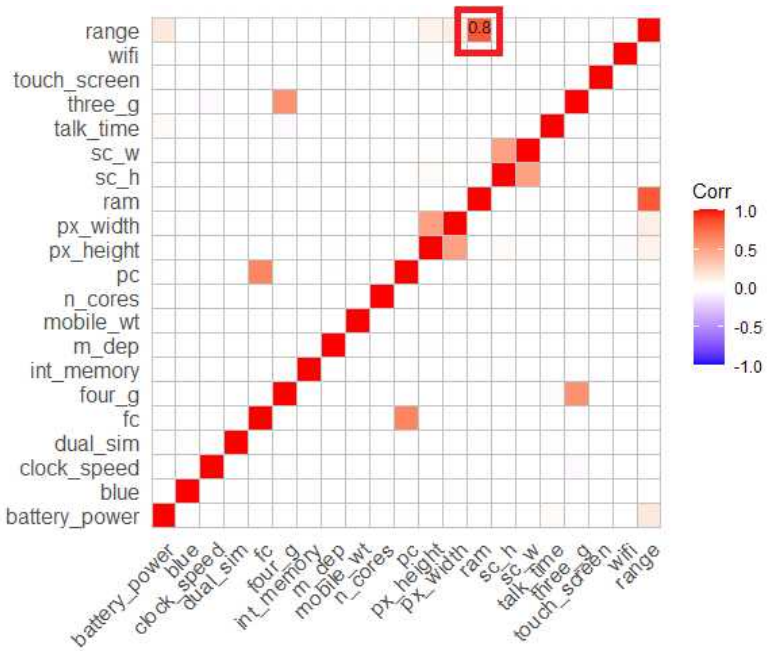
```
corrPrice <- price[,c(12,13,14,18)]
cols <- colorRampPalette(c("blue","skyblue"))
corrgram(corrPrice, col.regions=cols,order=TRUE, lower.panel=panel.fill,upper.panel=panel.pie,
         text.panel=panel.txt,main="Corrgram of Mobile Phone's X variables")
```



독립변수 (px_height, px_width, ram, three_g)간의 상관관계도표를 그려보았다.
상관관계도표는 여러 변수들에서 두 변수의 상관성을 볼 수 있는 표이지만, 상관성을 한눈에 알아볼 수 없고, 해석하기 위하여 하나하나 확인해야 한다는 단점이 존재한다. 개선점으로는 저렇게 panel_shade 또는 panel_fill의 경우 색깔의 음영 뿐만 아니라 색깔 위에 상관성

정도에 대한 수치 등을 기재한다면 더욱 좋은 도표가 될 수 있다. 또한 옆에 범례를 표시하면 좋을 것이다. 범례에는 어떤 색깔이 양의 상관관계인지, 또는 음의 상관관계인지를 알려주면 알아보기 쉬울 것이다. 추가적으로, 각 값이 어느 변수에 해당하는지 차트의 x축 y축처럼 기재되어 있으면 개선될 수 있다.

이렇게 개선한 상관관계도표를 library(ggcorrplot)을 통하여 실제로 적용해 보았다.



이렇게 표현한다면 변수 range가 변수 ram과 상관성이 0.8(임의로 부여한 수)이나 된다는 것을 금방 알아챌 수 있다. 또한 그 값이 양수인지, 음수인지를 범례를 통해 바로 알아챌 수 있다.

20개의 독립변수 모두를 로지스틱 회귀모델에 적용하면 에러가 발생한다. 따라서 유의수준이 낮은 값을 위주로 임의로 선택하여 모델을 생성하였다.

```
Call:
glm(formula = range ~ ., family = binomial(), data = priceTrain)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.14912  -0.03350   0.00057   0.05189   2.30960

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.205e+01  2.942e+00  -7.494 6.67e-14 ***
four_g      -9.250e-04  5.600e-01  -0.002   0.9987
n_cores      1.015e-01  9.495e-02   1.069   0.2849
px_height    3.295e-03  6.697e-04   4.920 8.64e-07 ***
px_width     2.737e-03  6.790e-04   4.031 5.55e-05 ***
ram          7.427e-03  8.959e-04   8.290  < 2e-16 ***
sc_h        -1.098e-01  5.240e-02  -2.096   0.0361 *
talk_time    8.321e-02  4.057e-02   2.051   0.0402 *
three_g      9.039e-01  6.657e-01   1.358   0.1745
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 831.11  on 599  degrees of freedom
Residual deviance: 145.27  on 591  degrees of freedom
AIC: 163.27

Number of Fisher Scoring iterations: 8
```

[로지스틱 회귀 모델 평가 기준]
: 선형회귀 분석때 쓰인 Coefficients 및 P-value, 이진분류기의 성능평가(Accuracy, Confusion Matrix, Precision, Recall, F1 measure)로 평가한다.

계수(Coefficient)는 선형회귀 모델과 동일한 방식으로 해석할 수 있다.
그러므로 Confusion Matrix로 평가를 진행한다.

2.4 로지스틱 회귀 결과해석 : 예측결과로 Confusion Matrix와 Recall, precision, F1 measure를 측정하고 설명하시오.

```
perf_eval <- function(cm){
  # true positive rate
  TPR = Recall = cm[2,2]/sum(cm[2,])
  # precision
  Precision = cm[2,2]/sum(cm[,2])
  # true negative rate
  TNR = cm[1,1]/sum(cm[1,])
  # accuracy
  ACC = sum(diag(cm)) / sum(cm)
  # balance corrected accuracy (geometric mean)
  BCR = sqrt(TPR*TNR)
  # f1 measure
  F1 = 2 * Recall * Precision / (Recall + Precision)

  re <- data.frame(TPR = TPR,
                   Precision = Precision,
                   TNR = TNR,
                   ACC = ACC,
                   BCR = BCR,
                   F1 = F1)
  return(re)
}
```

```
pred_prob <- predict(newPriceModel, priceTest, type="response")
pred_clas <- rep(0, nrow(priceTest))
pred_clas[pred_prob > 0.5] <- 1
cm <- table(pred=pred_clas, actual=priceTest$range)
```

```
> perf_eval(cm)
      TPR       Precision    TNR       ACC       BCR             F1
1 0.9139168 0.9218524 0.9231863 0.9185714 0.9185399 0.9178674
```

newPriceModel로 만든 pred_clas를 통해 임의로 정의한 함수 perf_eval(cm)을 통하여 나온 값을 통해 Confusion Matrix를 유추할 수 있다.

Confusion Matrix		Predicted	
		High(1)	Low(0)
Actual	High(1)	a	b
	Low(0)	c	d

1. 실제 High인 것 중 모델이 High으로 예측한 비율

Sensitivity (= Recall) = $\frac{a}{a+b}$ = 0.9139168

2. 실제 Low인 것 중 모델이 Low으로 예측한 비율

Specificity = $\frac{d}{c+d}$ = 0.9231863

3. 모델이 High이라 분류한 것 중 실제 High인 비율

Precision = $\frac{a}{a+c}$ = 0.9218524

4. 실제 High를 High이라 예측한 것과 실제 Low를 Low라 예측한 두 비율을 합한 비율

Accuracy = $\frac{a+d}{a+b+c+d}$ = 0.9185714

5. Recall과 Precision의 조화평균

F1 measure = $\frac{2*Recall*Precision}{Recall+Precision}$ = 0.9178674

여기서 Confusion Matrix를 통해
Recall = 0.9139168
Precision = 0.9218524
F1 measure = 0.9178674라는 값을 얻을 수 있다.

또한 정확도가 91.85% 정도로 핸드폰의 가격이 낮은 범주인지 높은 범주인지 예측할 수 있다.

2.5 (공통) 선형회귀와 로지스틱 회귀에 대해서 변수선택법을 사용하여 정확도를 높이시오.

적용할 변수선택법
1. 전진 선택법_Forward selection
: 예측 변수가 아무것도 추가되어 있지 않은 모델에서 개선 정도 예측이 가장 높은 변수를 하나씩 추가하면서 최선의 모델을 찾는 선택법
2. 변수 소거법_Backward Elimination
: 모든 예측 변수가 추가된 모델에서 예측 모델에 가장 영향이 없는 예측 변수를 하나씩 소거해 가면서 최선의 모델을 찾는 선택법
3. 단계적 선택법_Stepwise selection
: 모든 예측 변수가 추가된 모델에서 기준 통계치에 가장 도움이 되지 않는 변수를 삭제하거나, 모델에 빠져 있는 변수 중 기준 통계치를 가장 개선시키는 변수를 추가한다.

2.5 (선형회귀) 변수선택법을 사용하여 정확도를 높이시오.

```
student2Forward <- step(student2Model, direction = "forward") # 전진 선택법
student2Backward <- step(student2Model, direction = "backward") # 후진 소거법
student2Stepwise <- step(student2Model, direction = "both") # 단계적 선택법
```

이때, 같은 값을 보인 선택법들의 예측 변수를 확인하였다.

```
# 후진 선택법
school
age
activities
romantic
famrel
Walc
absences
G1
G2
```

```
# 단계적 선택법
school
age
activities
romantic
famrel
Walc
absences
G1
G2
```

두 선택법의 모델 변수들이 모두 동일한 것을 알 수 있다.

각 모델의 R-squared 값을 비교해보았다.

	R-squared
임의 선택	0.8351
전진 선택법	0.8435
후진 소거법	0.8382
단계적 선택법	0.8382

이를 통해서 Salary_Model에 대한 R-squared값이 가장 높은 변수선택법은 전진 선택법(Forward Selection)이며, 기존에 P-value값을 유의수준보다 낮은지 선택한 임의 선택의 R-squared 값인 0.8351보다 전진 선택법의 R-squared 값이 0.8435로 0.0084 정도 향상되었다.

다음은 전진 선택법(Forward Selection)의 모델에 설정된 예측 변수이다.

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.43655    2.05748  -0.212 0.832087
school      -0.43513    0.35338  -1.231 0.218993
sex         -0.14569    0.22634  -0.644 0.520206
age         -0.17471    0.09341  -1.870 0.062224 .
address      0.03986    0.26361   0.151 0.879882
famsize     -0.04739    0.21979  -0.216 0.829420
Pstatus     -0.16239    0.32746  -0.496 0.620268
Medu         0.12322    0.12122   1.017 0.310047
Fedu        -0.13206    0.11935  -1.106 0.269247
guardian     0.21148    0.22092   0.957 0.339057
traveltime   0.09914    0.15294   0.648 0.517243
studytime   -0.09205    0.13030  -0.706 0.480355
failures    -0.18136    0.15147  -1.197 0.231961
schoolsup    0.50058    0.31151   1.607 0.108926
famsup       0.18217    0.21881   0.833 0.405645
paid         0.07897    0.21500   0.367 0.713604
activities  -0.35701    0.20011  -1.784 0.075243 .
nursery     -0.23656    0.24682  -0.958 0.338466
higher       0.17891    0.48146   0.372 0.710402
internet    -0.19351    0.27619  -0.701 0.483962
romantic    -0.26473    0.21362  -1.239 0.216048
famrel       0.34910    0.11109   3.142 0.001812 **
freetime     0.06806    0.10621   0.641 0.522087
goout        0.01362    0.10294   0.132 0.894801
Dalc        -0.17763    0.14626  -1.214 0.225343
Walc         0.16247    0.11009   1.476 0.140880
health       0.06519    0.07131   0.914 0.361171
absences     0.04541    0.01294   3.508 0.000507 ***
G1           0.18952    0.05916   3.203 0.001479 **
G2           0.95769    0.05182  18.483  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.883 on 365 degrees of freedom
Multiple R-squared:  0.8435,  Adjusted R-squared:  0.8311
F-statistic: 67.85 on 29 and 365 DF,  p-value: < 2.2e-16
```

#제거하지 않은 모든 예측 변수가 추가된 모델임을 알 수 있다.

이를 통하여 독립변수의 개수가 많은 경우, R-squared만 고려할 것이 아닌, Adjusted R-squared 또한 고려해야 함을 알 수 있다.

2.5 (로지스틱 회귀) 변수선택법을 사용하여 정확도를 높이시오.

```
weather_fwd <- step(glm(RainTomorrow ~ 1,weather_train, family = binomial()),
                    direction = "forward", trace = 0, scpoe = formula(weather))
pred_prob <- predict(weather_fwd, weather_test, type="response")
pred_clas <- rep(0, nrow(weather_test))
pred_clas[pred_prob > 0.5] <- 1
cm <- table(pred=pred_clas, actual=weather_test$RainTomorrow)
perf_eval(cm) # 전진 선택법
```

```
priceBack <- step(glm(range ~ ., priceTest, family = binomial()), direction = "backward",
trace = 0, scope = list(lower=range ~ 1, upper=formula(priceModel)))
pred_prob <- predict(priceBack, priceTest, type="response")
pred_clas <- rep(0, nrow(priceTest))
pred_clas[pred_prob > 0.5] <- 1
cm <- table(pred=pred_clas, actual=priceTest$range)
perf_eval(cm) # 후진 소거법
```

```
> perf_eval(cm)
        TPR Precision       TNR       ACC       BCR        F1
1 0.9297218  0.918958 0.9218968 0.9257143 0.925801 0.9243086
```

```
priceStep <- step(glm(range ~ ., priceTest, family = binomial()), direction = "both", trace
= 0, scope = list(lower=range ~ 1, upper=formula(priceModel)))
pred_prob <- predict(priceStep, priceTest, type="response")
pred_clas <- rep(0, nrow(priceTest))
pred_clas[pred_prob > 0.5] <- 1
cm <- table(pred=pred_clas, actual=priceTest$range)
perf_eval(cm) # 단계적 선택법
```

```
> perf_eval(cm)
  TPR Precision TNR ACC BCR F1
1   1         1   1   1   1  1
```

각 모델에 대하여 Accuracy를 비교해보았다.

	정확도(Accuracy)
임의 선택	0.9185714
전진 선택법	NA
후진 소거법	0.9257143
단계적 선택법	1

여기서 전진 선택법은

이런 오류가 발생한다.
이를 통해서 price 모델에 대한 가장 높은 설명력을 보인 변수 선택법은 단계적 선택법(Stepwise Selection)이며, 기존에 P-value값을 유의수준보다 낮은 것을 선택한 임의 선택의 정확도인 0.9185714보다 단계적 선택법의 정확도 1로 0.0814286의 정확도를 더 높였다.

다음은 단계적 선택법(Stepwise Selection)의 모델에 설정된 예측 변수이다.

Coefficients:

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-48902.995	192697.227	-0.254	0.800
battery_power	8.932	35.189	0.254	0.800
blue	366.873	1475.253	0.249	0.804
dual_sim	-729.888	2895.931	-0.252	0.801
fc	-70.109	287.225	-0.244	0.807
int_memory	26.696	105.743	0.252	0.801
m_dep	-763.344	3039.843	-0.251	0.802
mobile_wt	-15.019	59.424	-0.253	0.800
n_cores	150.366	594.701	0.253	0.800
pc	84.394	336.767	0.251	0.802
px_height	5.124	20.212	0.254	0.800
px_width	5.041	19.850	0.254	0.800
ram	13.696	53.981	0.254	0.800
sc_h	-7.320	38.487	-0.190	0.849
three_g	228.051	943.504	0.242	0.809
touch_screen	-103.630	487.604	-0.213	0.832

마무리

지금까지 선형회귀 분석 및 로지스틱 분석을 통해 각각의 분석 모델을 설정하고, 모델이 얼마나 예측하고자 하는 값에 대해서 설명력을 갖는지 알아보았다.
두 분석을 위한 모델을 만들기 위해서 적합한 데이터셋을 찾고, 찾은 데이터의 전처리 작업을 진행하였다.
이외에도 여러 변수 선택법을 통해 다른 예측 변수를 가진 모델을 설정하여 이들의 정확도를 비교하였다.
이러한 작업을 통해 선형회귀와 로지스틱 회귀가 어떤 데이터셋의 분석에 적합한 것인지를 명확하게 구분할 수 있게 되었다.

* 참고자료
맨체스터 유나이티드 vs 레스터 시티 FC 조사 링크
(https://www.soccerbase.com/teams/head_to_head.sd?team_id=1724&team2_id=1527)
다중선형회귀분석 데이터셋 링크
(https://www.kaggle.com/dipam7/student-grade-prediction)
다중선형회귀분석 참고자료
(https://www.kaggle.com/hindelya/students-grade-prediction)
로지스틱회귀분석 데이터셋 링크
(https://www.kaggle.com/iabhishekofficial/mobile-price-classification?select=train.csv)
로지스틱회귀분석 참고자료
(https://www.kaggle.com/pierpaolo28/mobile-price-classification)

지금까지 데이터마이닝 과제#2입니다. 감사합니다.