2018.12.18

제조데이터 활용한

생산 불량 원인 분석

목차

EDA
(데이터의 구조와 특징 파악하기)
- 이상치/결측치 확인하기
- 데이터 준비하기
- 시각화

Machine Learning
(예측 알고리즘 만들기)
- 회귀분석
- 분류분석

Estimation
(모델 평가하기)
- 각 모델의 정확도 비교하여 최적 모델을 선정하기

EDA: 데이터 구조와 특징 파악하기

prod_date	prod_no	prod_nam	degree	mold	prod	s_no	fix_time	a_speed	b_speed	separation	s_separatic	rate_terms	mpa	load_time	highpressu	c_thickness
2014-03-0	45231-3B4	Oil Gasket	1	양산	생산	349892	68.3	0.492	1.702	206.1	712.1	83	29.6	0	96	32.9
2014-04-0	90784-76C	Oil Gasket	2	생산대기	생산	871869	73.1	0.469	1.561	246.6	652.8	80	28.9	0	90	40.1
2014-04-0	90784-76C	Oil Gasket	2	생산대기	생산	872996	76.3	0.461	1.539	156.9	741.9	88	29.2	0	82	27
2014-03-3	90784-76C	Oil Gasket	1	승인대기	생산	368872	65.4	0.566	1.551	157.9	735	82	29.8	0	81	54.2
2014-03-0	90784-76C	Oil Gasket	2	생산대기	생산	350083	81.5	0.501	1.593	182.9	715.3	82	29.4	0	80	38.2
2014-03-2	45231-3B6	Oil Gasket	2	양산	생산	359147	67.6	0.502	1.499	195.2	722.8	90	30.9	0	80	36.5
2014-03-0	90784-76C	Oil Gasket	2	생산대기	생산	350080	63.3	0.497	1.593	182.9	715.3	81	29.5	0	80	35
2014-03-3	45231-3B6	Oil Gasket	1	양산	생산	368671	82.1	0.538	1.567	193.8	698.1	86	29.4	0	79	30.3
2014-04-3	90784-76C	Oil Gasket	2	생산대기	생산	892322	71	0.477	1.638	244.8	655.5	91	28.5	0	78	34.6

autopart.csv 데이터는

자동차 부품의 생산데이터로서 공정변수들과 함께 생산된 oil gasket의 탕구 두께 기록된 자료이다.

우리는 생산품 품질(탕구 두께)에 주된 영향을 미치는 공정변수 및 그 연관성 등을 파악하여 본다.

EDA: 데이터 구조와 특징 파악하기

```
> str(car)
'data.frame':   34127 obs. of  17 variables:
 $ prod_date         : Factor w/ 34127 levels "2014-03-01 오전 10:00:33",..: 5918 5917 27081 26236 20023 13062 5922 25371 5370
 ...
 $ prod_no           : Factor w/ 6 levels "45231-3B400",..: 5 5 6 6 4 2 5 6 6 6 ...
 $ prod_name         : Factor w/ 1 level "Oil Gasket": 1 1 1 1 1 1 1 1 1 1 ...
 $ degree            : int  2 2 2 2 1 2 2 2 2 2 ...
 $ mold              : Factor w/ 3 levels "a","b","c": 3 3 1 1 3 3 3 1 1 1 ...
 $ prod              : Factor w/ 1 level "생산": 1 1 1 1 1 1 1 1 1 1 ...
 $ s_no              : int  351134 351133 873570 872723 365279 358252 351138 871491 350457 871490 ...
 $ fix_time          : num  65.9 80.8 81.1 76 76.5 67.2 82.7 85 81 85.3 ...
 $ a_speed           : num  0.471 0.475 0.495 0.46 0.516 0.498 0.473 0.595 0.661 0.607 ...
 $ b_speed           : num  1.7 1.7 1.51 1.68 1.49 ...
 $ separation        : num  185 185 157 251 185 ...
 $ s_separation      : num  713 713 743 652 717 ...
 $ rate_terms        : int  84 84 83 80 77 85 82 80 86 81 ...
 $ mpa               : num  29.1 29.1 28.7 28.8 29.4 30.4 29.1 26.1 27.2 26.1 ...
 $ load_time         : num  19.6 0 0 0 0 0 19.6 18.1 19.1 18.1 ...
 $ highpressure_time: int  73 73 60 58 76 73 73 65534 70 65534 ...
 $ c_thickness       : num  66 65.6 65.6 60.4 58.2 56.6 55.7 55.3 55.2 55.2 ...
```

autoparts 데이터는

34117개의 관측치와 15개의 특징들로 구성되어 있습니다.

반응변수는 (c_thickness)이며,

7개의 예측변수(fix_time, a_speed, b_speed, rate_terms, mpa, load_time, highpressure_time)을 사용해 분석할 것입니다. (예측변수 설정은 데이터 설명에 따름)

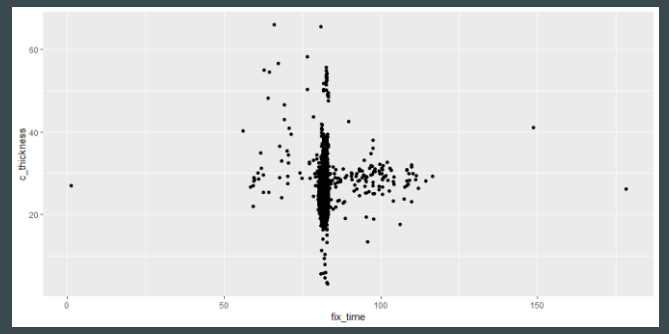
단일 level인 'prod_no', 'prod'특징은 분석에 사용하지 않을 것이다.

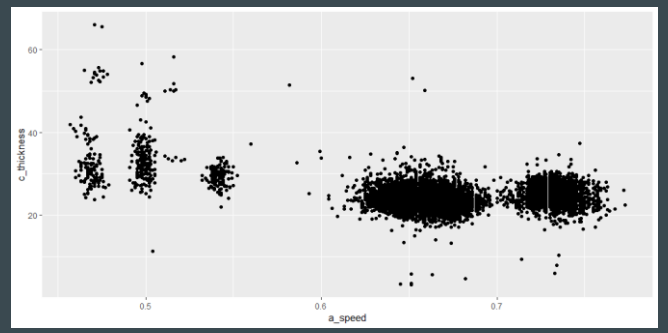
EDA: 데이터 구조와 특징 파악하기

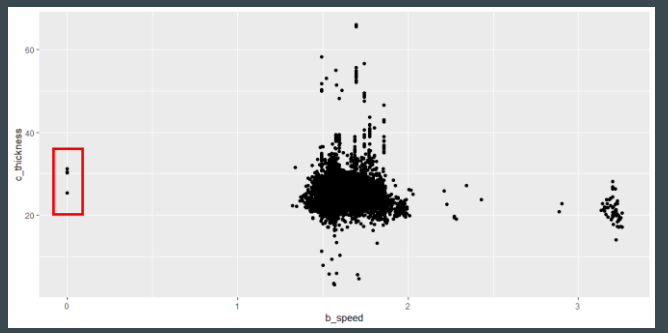
```
> summary(car)
                  prod_date          prod_no          prod_name        degree      mold        prod
 2014-03-01 오전 10:00:33:    1   45231-3B400 : 3434   Oil Gasket:34127   Min.   :1.000   a:19917   생산:34127
 2014-03-01 오전 10:01:56:    1   45231-3B610 : 2378                      1st Qu.:1.000   b: 1850
 2014-03-01 오전 10:03:18:    1   45231-3B641 :  929                      Median :2.000   c:12360
 2014-03-01 오전 10:04:41:    1   45231-3B660 : 5550                      Mean   :1.655
 2014-03-01 오전 10:06:03:    1   45231-P3B750:   69                      3rd Qu.:2.000
 2014-03-01 오전 10:07:27:    1   90784-76001 :21767                      Max.   :2.000
 (Other)                  :34121
      s_no           fix_time          a_speed          b_speed        separation      s_separation      rate_terms
 Min.   :345142   Min.   :  1.00   Min.   :0.4570   Min.   :0.000   Min.   :  0.0   Min.   :  0.0   Min.   :76.00
 1st Qu.:353801   1st Qu.: 81.00   1st Qu.:0.6020   1st Qu.:1.576   1st Qu.:183.6   1st Qu.:657.1   1st Qu.:83.00
 Median :362403   Median : 81.90   Median :0.6480   Median :1.633   Median :187.9   Median :711.1   Median :86.00
 Mean   :511816   Mean   : 82.66   Mean   :0.6359   Mean   :1.642   Mean   :202.7   Mean   :696.6   Mean   :85.73
 3rd Qu.:871791   3rd Qu.: 84.90   3rd Qu.:0.6610   3rd Qu.:1.684   3rd Qu.:243.2   3rd Qu.:715.3   3rd Qu.:88.00
 Max.   :892890   Max.   :178.30   Max.   :0.8080   Max.   :3.257   Max.   :294.5   Max.   :748.9   Max.   :97.00

      mpa           load_time      highpressure_time   c_thickness
 Min.   :24.80   Min.   : 0.00   Min.   :    0.00   Min.   : 0.30
 1st Qu.:75.00   1st Qu.:18.20   1st Qu.:   64.00   1st Qu.:22.20
 Median :76.10   Median :19.20   Median :   70.00   Median :23.90
 Mean   :74.03   Mean   :19.09   Mean   :   87.51   Mean   :24.06
 3rd Qu.:77.80   3rd Qu.:19.70   3rd Qu.:   73.00   3rd Qu.:25.50
 Max.   :82.10   Max.   :23.40   Max.   :65534.00   Max.   :66.00
```

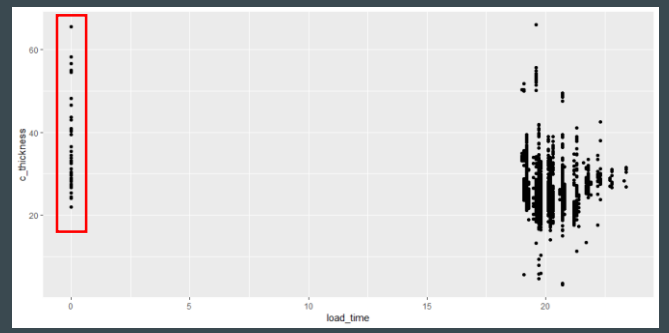
- ‘prod_date’ : 생산일자
- ‘prod_no’ : 생산품의 생산번호
- ‘prod_name’ : 생산품의 이름
- ‘degree’ : ? (데이터 설명 없음)
- ‘mold’ : ‘생산대기(a)’, ‘승인대기(b)’, ‘양산(c)’의 상태를 나타냄
- ‘prod’ : 생산
- ‘s_no’ : 생산번호

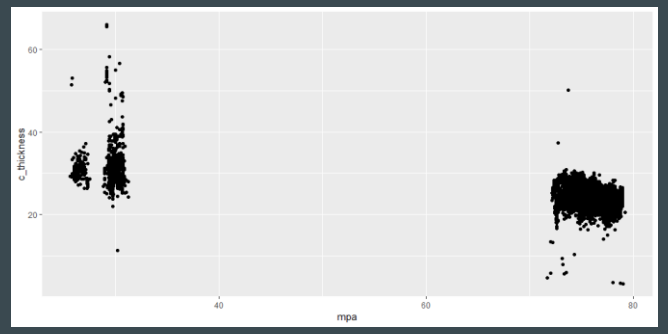
EDA: 이상치와 결측치 확인하기(산점도)

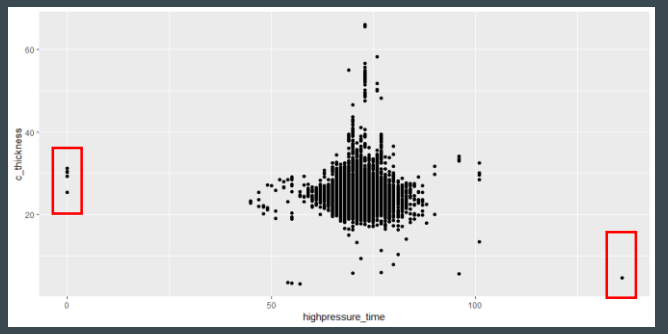


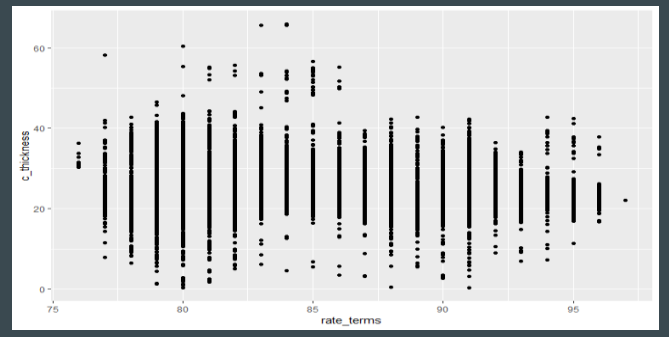






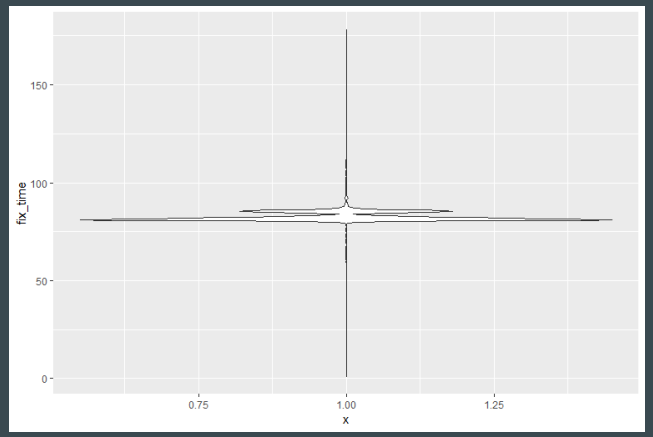


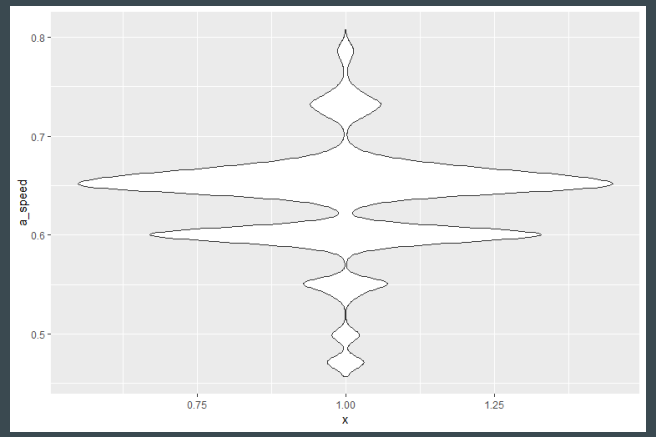


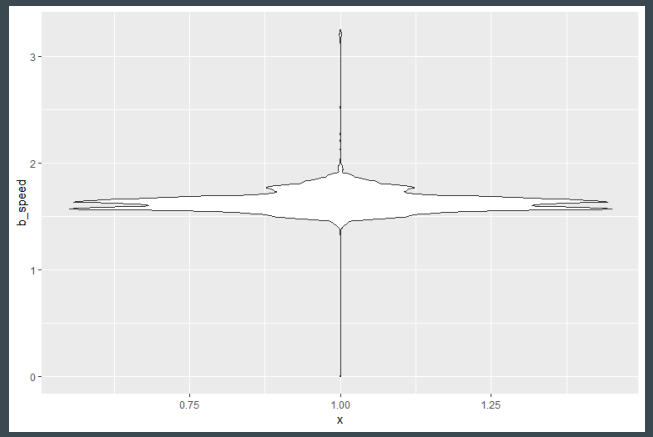


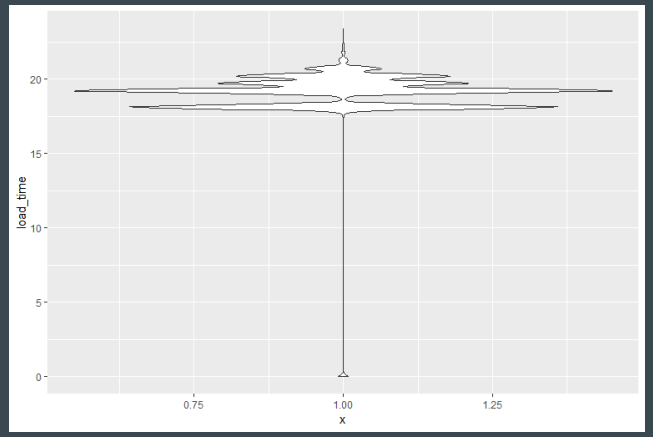
반응변수인 c_thickness과의 산점도에서 군집과 떨어진 점들이 관측된다.
B_speed, highpressure_time, load_time의 경우에는 이상치로,
Mpa의 경우에는 새로운 군집으로 보여진다.

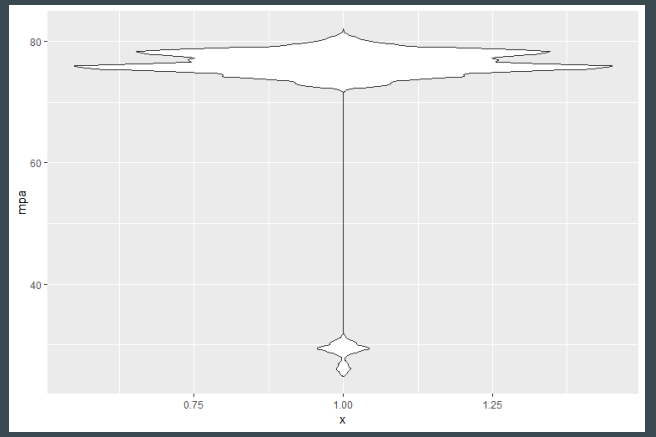
EDA: 이상치와 결측치 확인하기(상자그림)

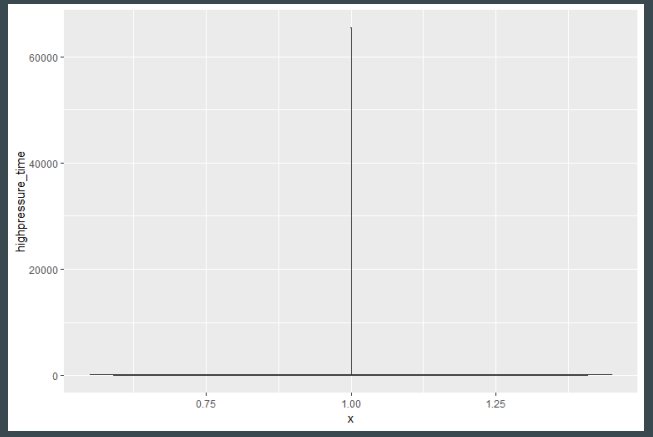












꼬리가 긴 모양을 가진 상자그림은 이상치를 나타낸다.

fix_time, b_speed, load_time, highpressure_time에서 이상치가 있을 것이라 예상된다.

EDA: 이상치와 결측치 제거하기

이상치를 확인한 후, 이상치 행을 삭제한다.

```
   fix_time
Min.    :  1.00
1st Qu.:  81.00
Median :  81.90
Mean    :  82.66
3rd Qu.:  84.90
Max.    :178.30
```

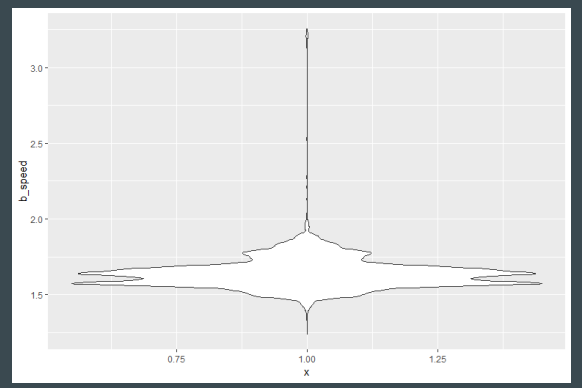
```
    b_speed
Min.    :0.000
1st Qu.:1.576
Median :1.633
Mean    :1.642
3rd Qu.:1.684
Max.    :3.257
```

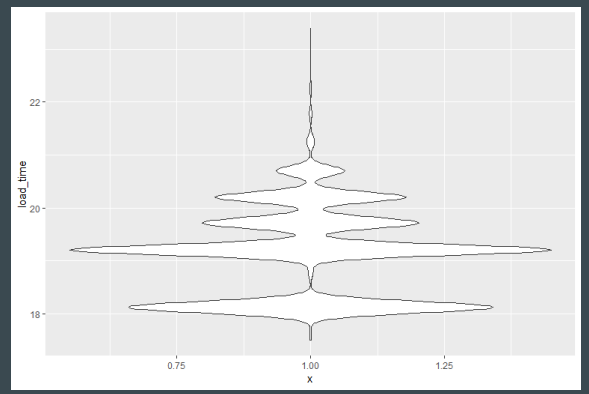
```
   load_time
Min.    : 0.00
1st Qu.:18.20
Median :19.20
Mean    :19.09
3rd Qu.:19.70
Max.    :23.40
```

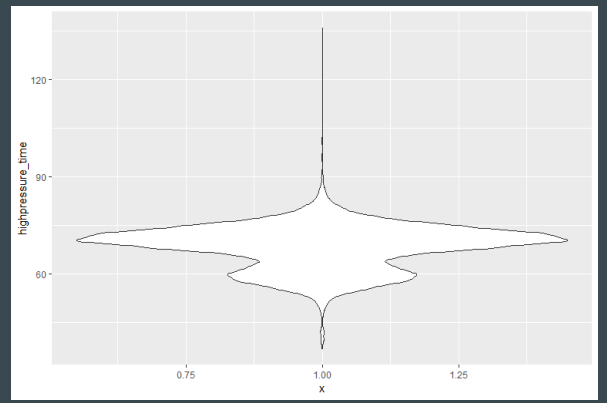
```
highpressure_time
Min.    :    0.00
1st Qu.:   64.00
Median :   70.00
Mean    :   87.51
3rd Qu.:   73.00
Max.    :65534.00
```

```r
car1 <- car1[car1$separation != 0,] ## 34139->34135
car1 <- car1[(car1$highpressure_time != 0)*(car1$highpressure_time != 65534.00) == 1,] ## 34135->34124
car1 <- car1[car1$load_time != 0,] ##34124->33927
```

이상치 정리 후, 상자그림







EDA: 이상치와 결측치 제거하기

이상치를 확인한 후, 이상치 행을 삭제한다.

```
   fix_time
Min.    :  1.00
1st Qu.:  81.00
Median :  81.90
Mean    :  82.66
3rd Qu.:  84.90
Max.    :178.30
```

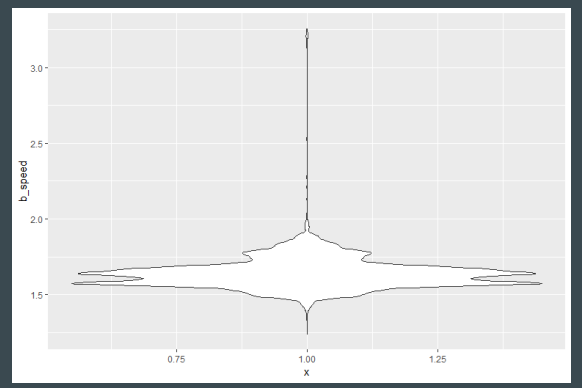
```
    b_speed
Min.    :0.000
1st Qu.:1.576
Median :1.633
Mean    :1.642
3rd Qu.:1.684
Max.    :3.257
```

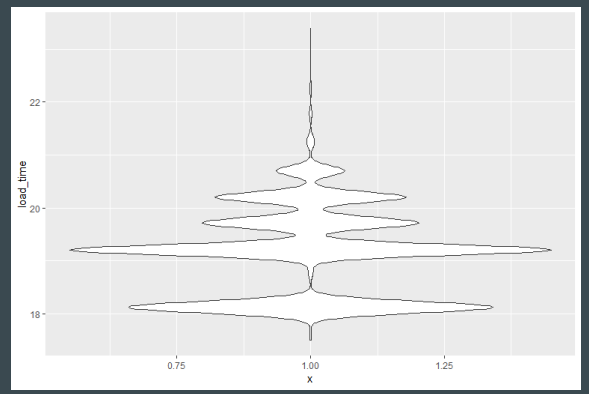
```
   load_time
Min.    : 0.00
1st Qu.:18.20
Median :19.20
Mean    :19.09
3rd Qu.:19.70
Max.    :23.40
```

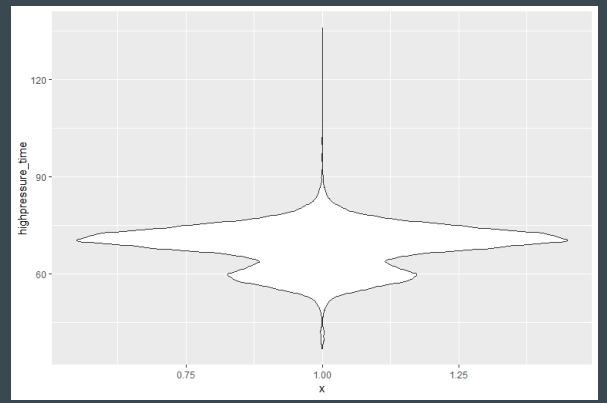
```
highpressure_time
Min.    :    0.00
1st Qu.:   64.00
Median :   70.00
Mean    :   87.51
3rd Qu.:   73.00
Max.    :65534.00
```

```r
car1 <- car1[car1$separation != 0,] ## 34139->34135
car1 <- car1[(car1$highpressure_time != 0)*(car1$highpressure_time != 65534.00) == 1,] ## 34135->34124
car1 <- car1[car1$load_time != 0,] ##34124->33927
```

이상치 정리 후, 상자그림







EDA: 데이터 준비하기

- 프로젝트 목적에 맞게 mold가 '양산'인 데이터만 추출하여 분석한다.

```
car0<-filter(car, mold == "양산")
car0$mold <- NULL #사용한 column은 삭제한다.
```

- 'prod'와 'prod_name' column은 factor가 1 level이기 때문에 분석에서 제외한다.

```
car0$prod <- NULL
car0$prod_name <- NULL
```

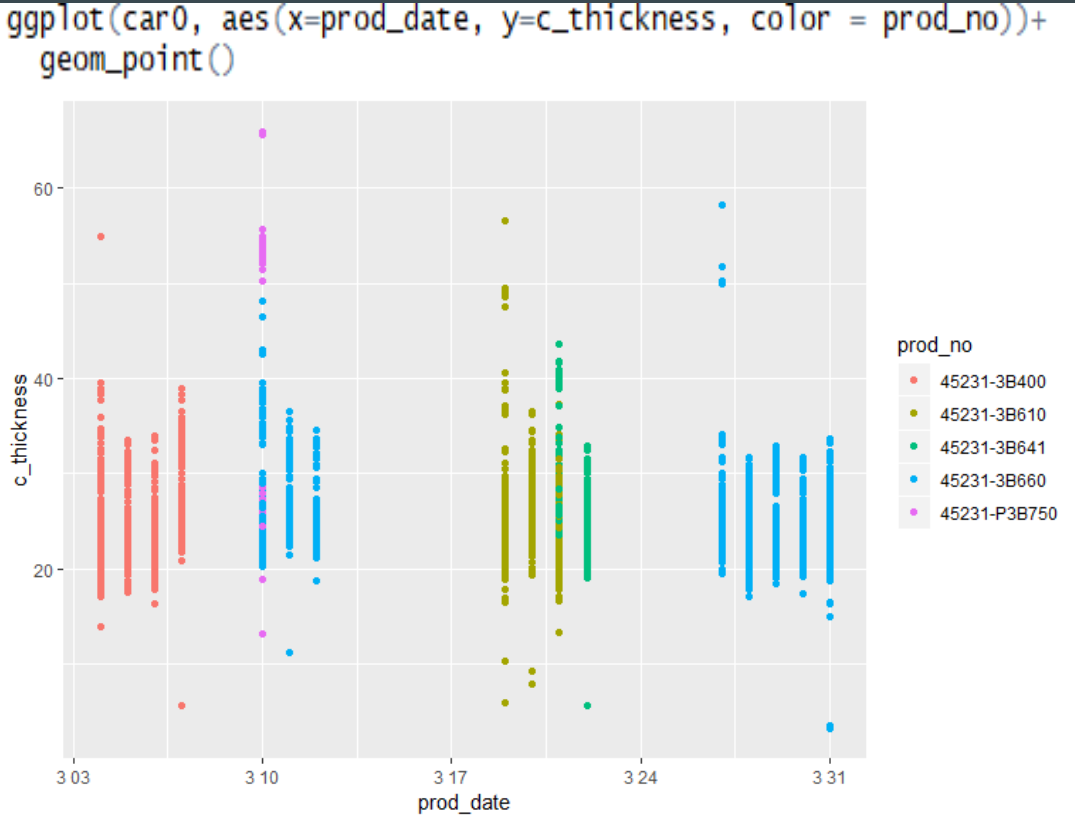
- Factor형 특징들을 시각화하여 데이터 셋을 준비합니다.
  이를 위해 'prod_date'의 자료형을 factor->date로 변환시켜준다.

```
car0$prod_date <- as.Date(car0$prod_date, format = "%Y-%m-%d") #날짜형식 변환(factor -> Date)
```

EDA: 데이터 준비(car0)하기

- Factor형 특징들을 시각화하여 데이터 셋을 준비합니다
  c_thickness//prod_date와 prod_no의 관계

```
ggplot(car0, aes(x=prod_date, y=c_thickness, color = prod_no))+
  geom_point()
```



→ 제품번호 45231-3B400에 해당하는
oil gasket에 대한 분석을 실시한다.
(생산 날짜와 데이터의 분포를 기준으로 선택하였음,
생산 번호에 따라 공정이 다르게 변할 수 있으므로)

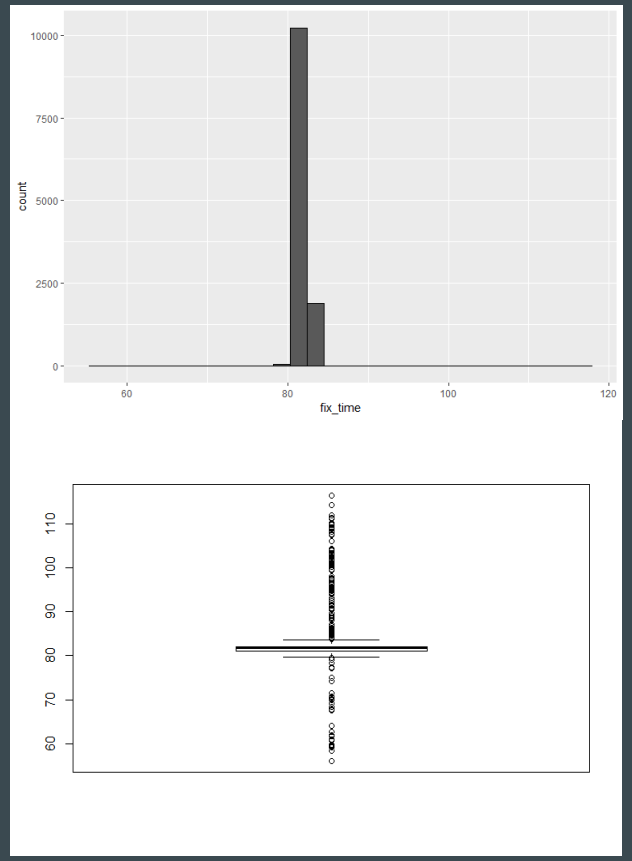
```
car0 <- car0[car0$prod_no == "45231-3B400",]
```

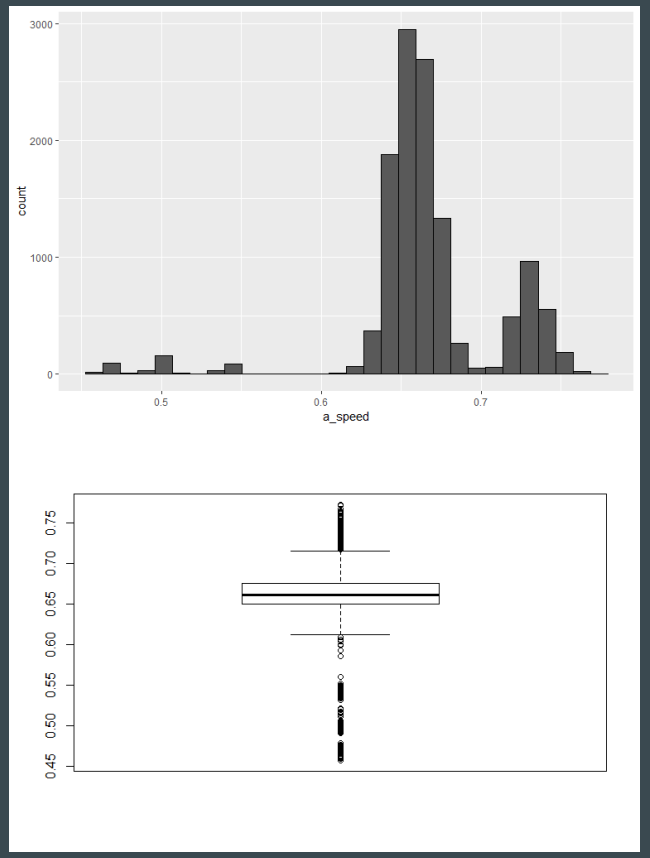
→ 탕구두께에 영향을 줄 것으로 생각되는 공정변수로
fix_time, a_speed, b_speed,
mpa, load_time, highpressure_time
으로 데이터 셋 car1을 만든다.

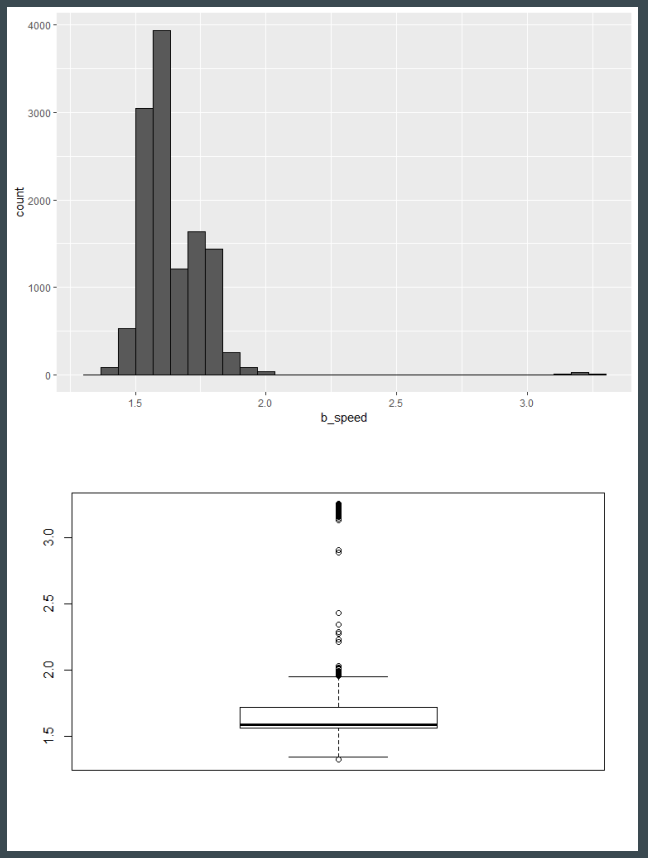
```
> head(car1)
  fix_time a_speed b_speed separation s_separation rate_terms  mpa load_time highpressure_time c_thickness
1     95.7   0.647   1.582      233.5        679.1         91 72.0      21.7               101        13.3
2     82.1   0.748   1.536      171.8        725.6         91 26.3      19.7               101        28.5
3     81.8   0.505   1.582      233.5        679.1         91 30.1      19.7               101        29.5
4     81.9   0.503   1.582      233.5        679.1         91 30.0      19.7               101        29.6
5     81.8   0.502   1.582      233.5        679.1         90 30.0      19.7               101        30.0
6     82.1   0.498   1.582      233.5        679.1         90 29.8      19.7               101        32.5
```

EDA: 시각화하기

(fix_time, a_speed, b_speed)

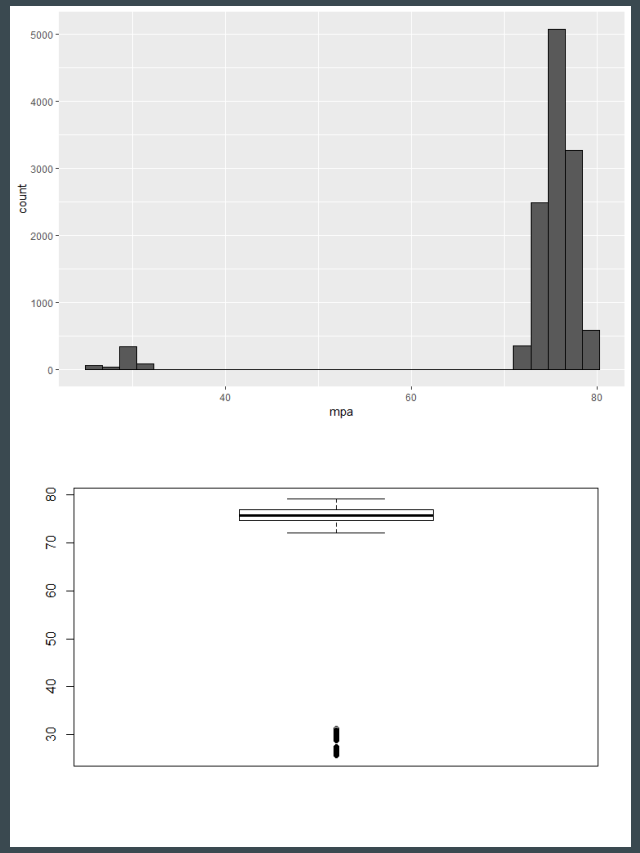


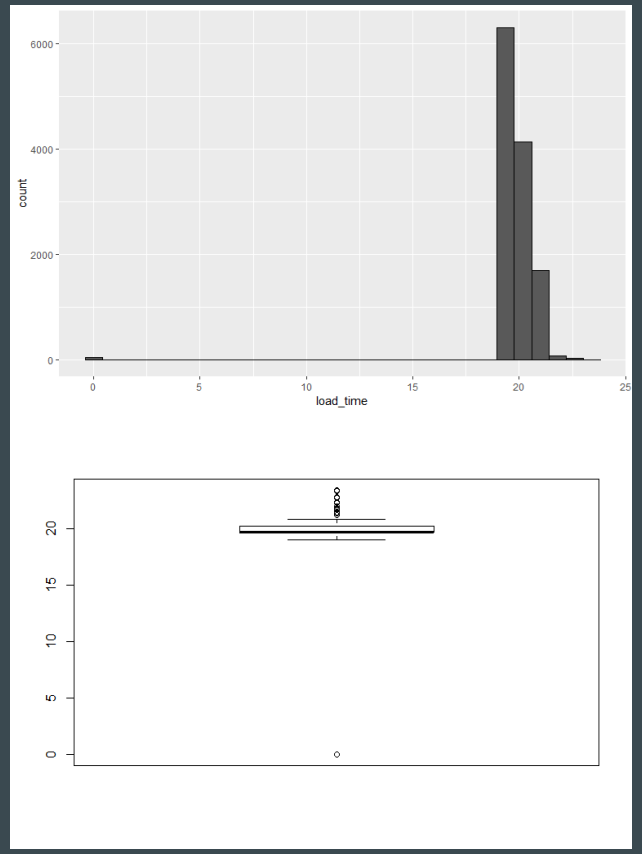


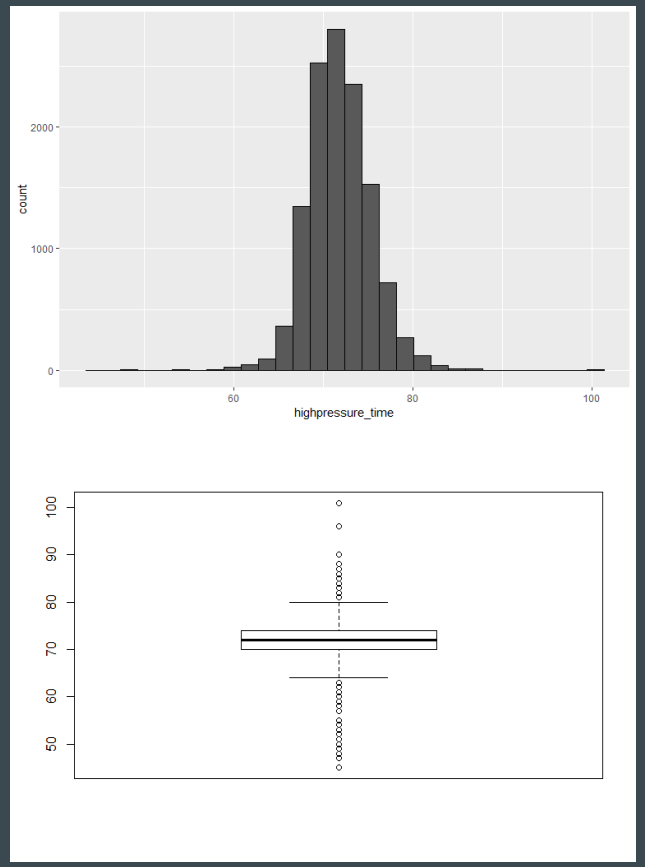


EDA: 시각화하기

(mpa, load_time, highpressure_time)







EDA: 시각화하기

변수 시각화를 통해

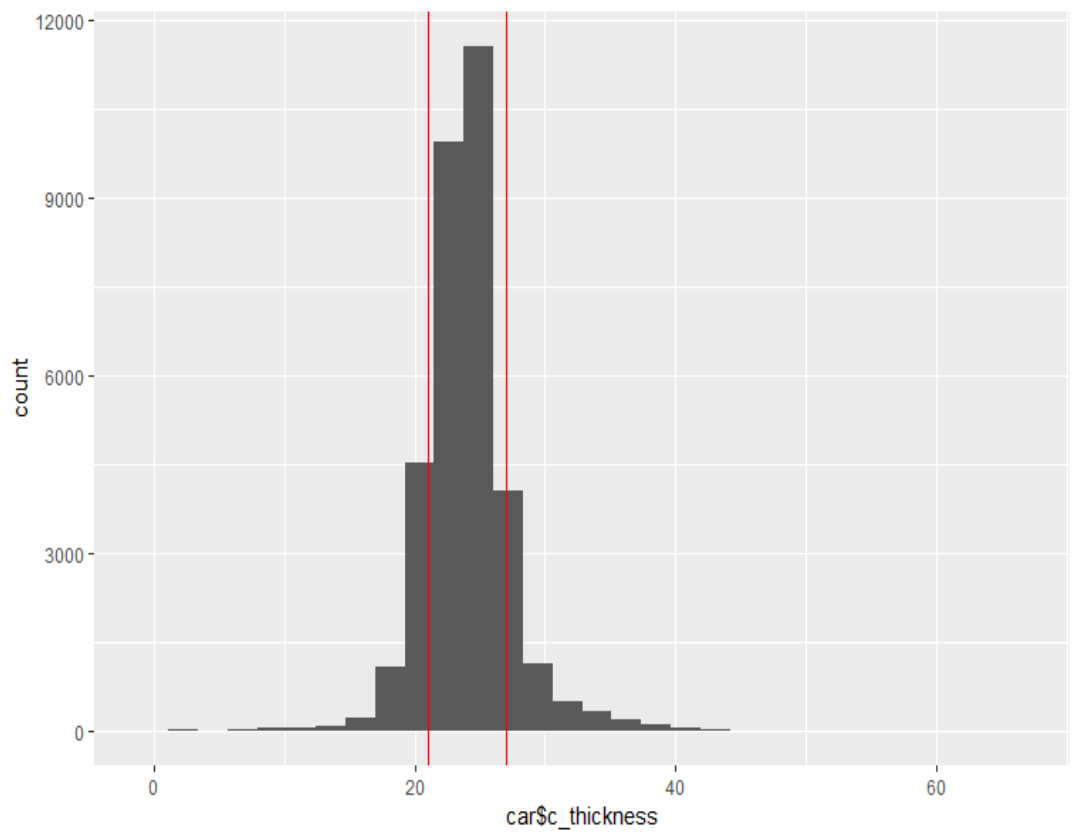
fix_time은 대부분이 80~90 사이에 몰려 있음을 알 수 있다.

a_speed, b_speed, mpa, road_time, highpressure_time의 값이
특정구간에 집중되는 경향이 있으나 일부 자료들은 분포의 중심에서 떨어져 있다.

특히 대부분 공정변수의 값들이 크게 두 그룹으로 분리되어 있는 것처럼 보인다.

작은 그룹에서 불량이 나올 것으로 예상된다.

EDA: 시각화하기 (c_thickness)



c_thickness의 시각화를 통해

자료의 범위를 크게 벗어나는 점들(10 미만, 50 초과)이 탐지된다.
이는 명백한 불량으로 볼 수 있으며
불량이 나타나는 프로세스 분석시 의미있는 정보를 제공할 수도 있으나
통계적 모형화를 실시할 때 지나치게 큰 영향력을 발휘할 수 있다고
판단되므로 제거하고 분석할 것이다.

```
car0 <- car0[(car0$c_thickness > 10)*(car0$c_thickness < 50) ==1,]
```

```
  c_thickness
 Min.   : 0.30
 1st Qu.:22.20
 Median :23.90
 Mean   :24.06
 3rd Qu.:25.50
 Max.   :66.00
```

```
  c_thickness
 Min.   :10.30
 1st Qu.:22.80
 Median :24.10
 Mean   :24.39
 3rd Qu.:25.60
 Max.   :49.90
```

EDA: 시각화하기 (c_thickness)

```
> describe(car0$c_thickness)
   vars     n  mean   sd median trimmed  mad  min  max range skew kurtosis   se
X1    1 12319 24.39 2.64   24.1    24.2 2.08 10.3 49.9  39.6 1.77     9.77 0.02
```

C_thickness의 describe함수를 통해

평균 24.240이며, 히스토그램에서 비교적 두꺼운 제품이 많이 발생하는 것을 볼 수 있다.

왜도(skewness)가 1.71로 양수의 값을 가지므로,
분포가 대칭이 아니라 오른쪽으로 약간 긴 꼬리를 가질 것이다.

즉, 불량 발생시 두께가 얇은 쪽 보다는 두꺼운 쪽으로 발생할 가능성이 높다고 여겨진다.

EDA: 시각화하기

공정변수의 영향력 분석을 위해
공정변수의 수준에 따른 탕구두께의 분포를 시각화를 할 것이다.

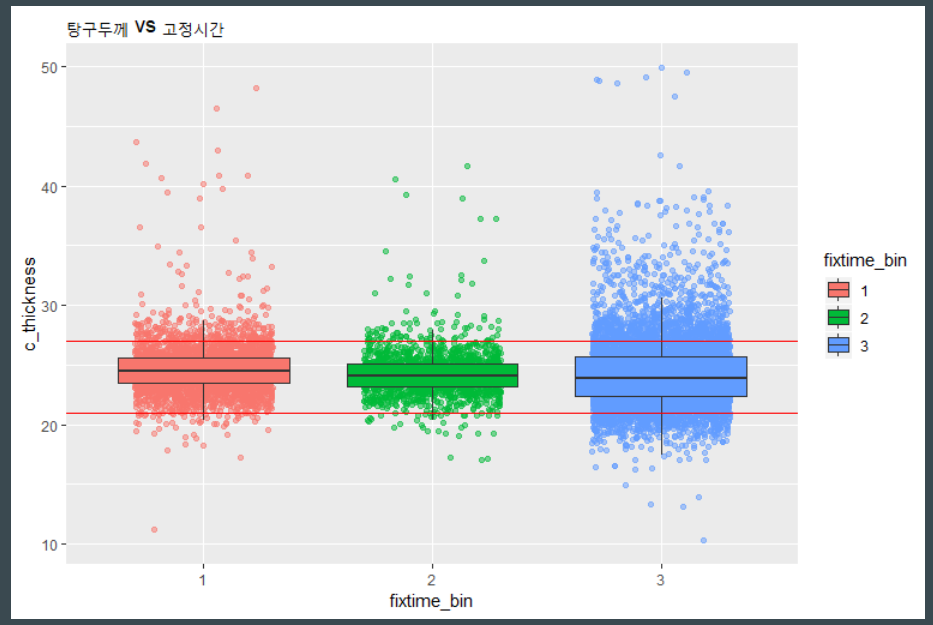
```
   fix_time        a_speed          b_speed        separation     s_separation     rate_terms         mpa
Min.   : 56.00   Min.   :0.4570   Min.   :1.323   Min.   :145.4   Min.   :672.6   Min.   :78.00   Min.   :25.70
1st Qu.: 81.10   1st Qu.:0.6500   1st Qu.:1.563   1st Qu.:176.9   1st Qu.:708.5   1st Qu.:85.00   1st Qu.:74.60
Median : 81.80   Median :0.6610   Median :1.589   Median :184.0   Median :714.1   Median :87.00   Median :75.70
Mean   : 81.82   Mean   :0.6662   Mean   :1.640   Mean   :181.9   Mean   :715.7   Mean   :87.83   Mean   :73.83
3rd Qu.: 82.10   3rd Qu.:0.6760   3rd Qu.:1.719   3rd Qu.:188.9   3rd Qu.:723.1   3rd Qu.:92.00   3rd Qu.:76.90
Max.   :116.50   Max.   :0.7730   Max.   :3.257   Max.   :235.3   Max.   :748.9   Max.   :95.00   Max.   :79.20
  load_time     highpressure_time  c_thickness
Min.   : 0.00   Min.   : 45.00    Min.   :10.30
1st Qu.:19.70   1st Qu.: 70.00    1st Qu.:22.80
Median :19.70   Median : 72.00    Median :24.10
Mean   :19.83   Mean   : 71.91    Mean   :24.39
3rd Qu.:20.20   3rd Qu.: 74.00    3rd Qu.:25.60
Max.   :23.40   Max.   :101.00    Max.   :49.90
```

```
car1$fixtime_bin <- as.factor((car1$fix_time > 81.3) + (car1$fix_time > 81.1) + 1)
car1$as_bin <- as.factor((car1$a_speed > 0.659) + (car1$a_speed > 0.667) + 1)
car1$bs_bin <- as.factor((car1$b_speed > 1.632) + (car1$b_speed > 1.732) + 1)
car1$mpa_bin <- as.factor((car1$mpa > 76.4) + (car1$mpa > 75.4) + 1)
car1$loadtime_bin <- as.factor((car1$load_time > 20.1) + (car1$load_time > 20.3) + 1)
car1$hp_bin <- as.factor((car1$highpressure_time > 72) + (car1$highpressure_time > 69) + 1)
```

변수들은 모두 연속형변수이기에 더 나은 시각화를 위해
각 변수들을 임의로 3개의 구간(low,middle,high = 1,2,3)으로 나누어
각 구간별로 탕구두께에 대한 상자그림을 출력한다.

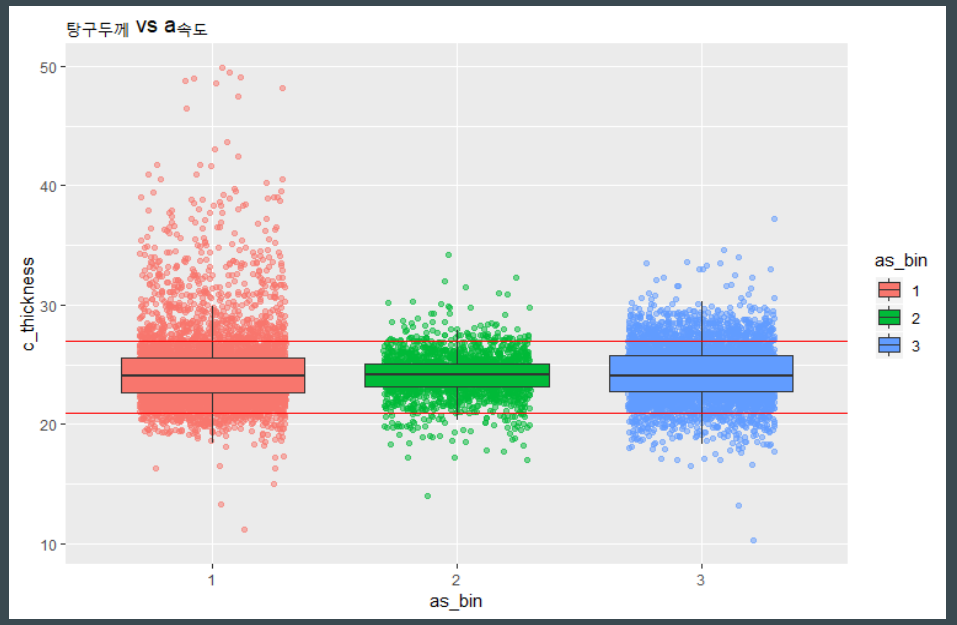
EDA: 시각화하기

1. fix_time



→ fix_time이 보통 수준보다 길면,
탕구두께의 변동도 커진다.

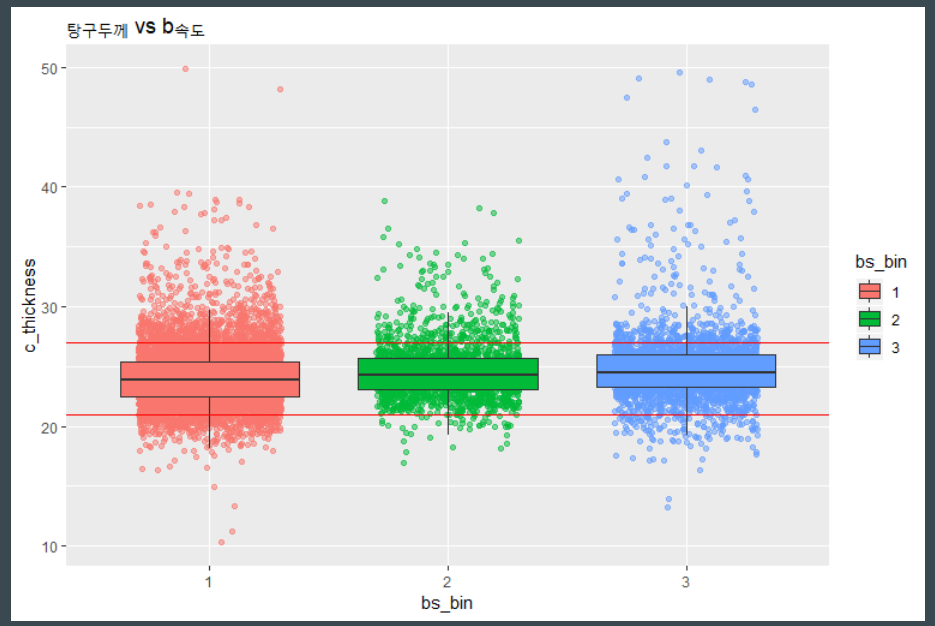
2. a_speed



→ a_speed이 낮은 수준이면,
탕구두께의 변동도 커진다.

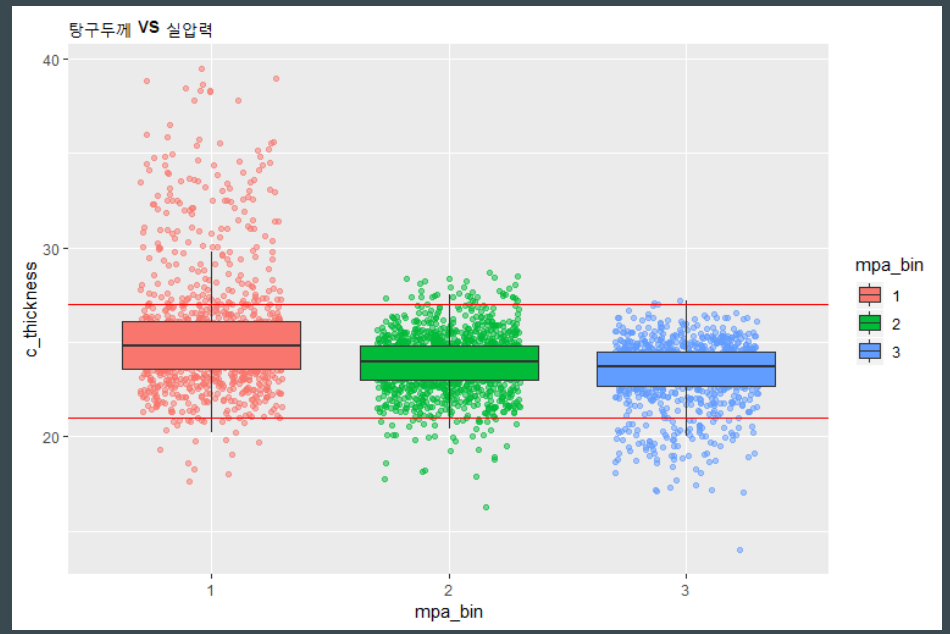
EDA: 시각화하기

3. b_speed



→ b_speed은 뚜렷한 경향성은 관측되지 않는다.

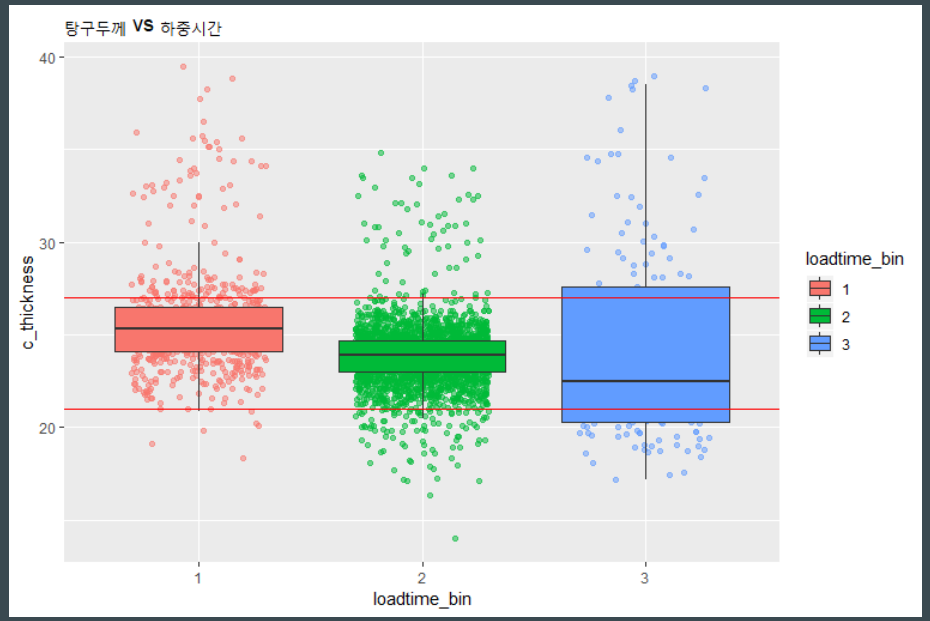
4. mpa



→ 실압력이 낮은 수준이면,
탕구두께가 두꺼워지는 경향이 있고.
→ 실압력이 높은 수준이면
탕구두께는 얇아지는 경향을 보인다.

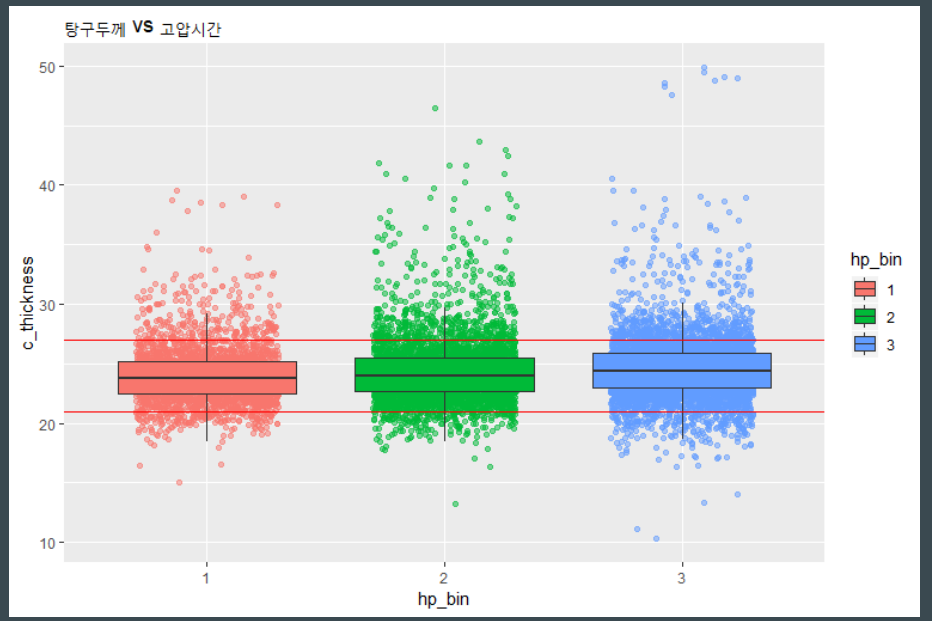
EDA: 시각화하기

5. load_time



→ 하중시간이 길수록
두께가 전반적으로 얇아지는 경향이 보이며,
→ 특히 하중시간이 가장 높은 수준에 속할 때는
두께의 변동이 매우 크게 증가하는 것으로 보인다.

6. highpressure_time

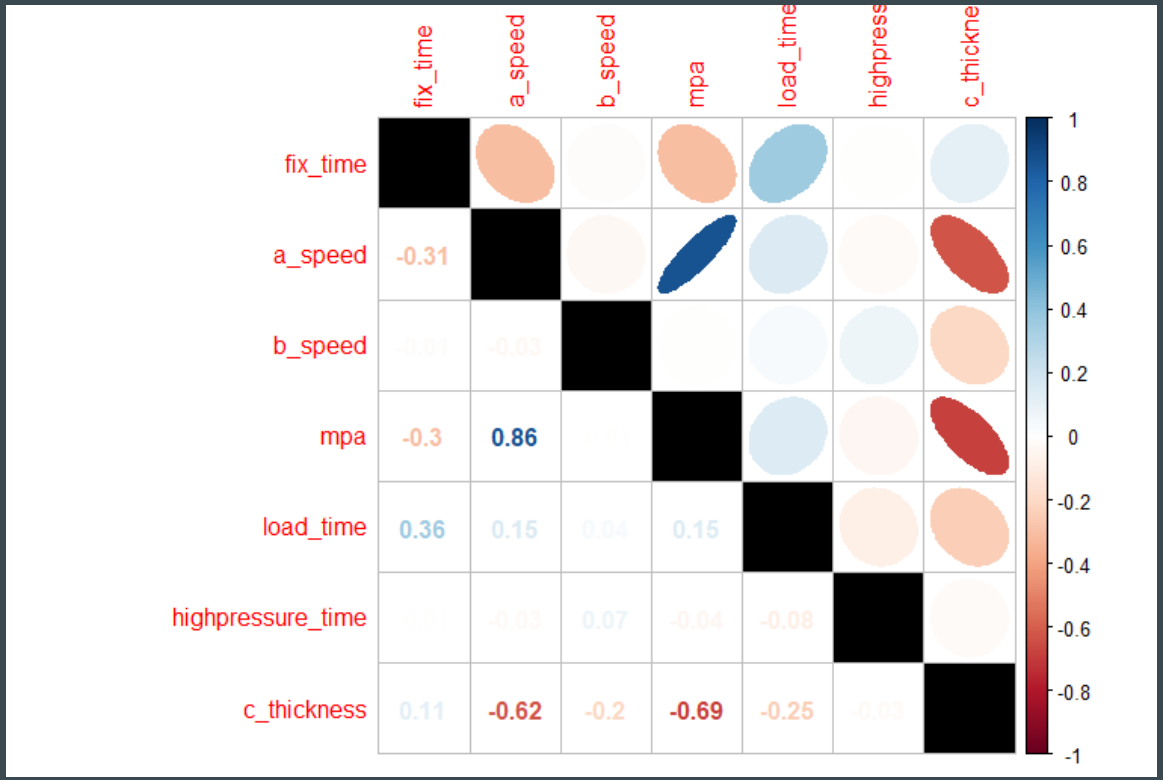


→ 고압시간의 경우 뚜렷한 경향성은 관측되지 않는다.

M/L: 회귀분석(상관분석)

회귀분석 전, 탕구두께 및 공정변수들간의 상관계수를 계산하여,
선형적 연관성의 정도를 파악한다.

```
corrplot.mixed(cor(car1),  upper = "ellipse" ,lower="number",tl.pos="lt",bg="black")
```



→ mpa과 a_speed 간의 상관관계가 가장 강하게 나타남.

→ c_thickness와 가장 강한 선형적 연관성을 가지는 변수는 a_speed와 mpa으로 파악되며, 모두 음의 상관이다.

→ a_speed와 mpa이 커질수록, c_thickness는 얇아지는 경향성이 있는 것으로 파악된다.

M/L: 회귀분석
1.선형회귀모형(Linear Model)

```
lm1 <- lm(c_thickness ~ .,data=car1)
```

```
Call:
lm(formula = c_thickness ~ ., data = car1)

Residuals:
    Min      1Q  Median      3Q     Max
-9.6629 -0.8977  0.0934  0.9760  6.9344

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)      59.295028   1.721474  34.444  < 2e-16 ***
fix_time         -0.092813   0.017914  -5.181 2.34e-07 ***
a_speed          -9.862596   1.547621  -6.373 2.10e-10 ***
b_speed          -2.333781   0.133943 -17.424  < 2e-16 ***
mpa              -0.140038   0.005618 -24.927  < 2e-16 ***
load_time        -0.219416   0.025268  -8.684  < 2e-16 ***
highpressure_time -0.031957   0.007375  -4.333 1.51e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.614 on 3424 degrees of freedom
Multiple R-squared:  0.5446,    Adjusted R-squared:  0.5438
F-statistic: 682.4 on 6 and 3424 DF,  p-value: < 2.2e-16
```

→선형회귀모형 적합결과 모든 공정변수들이 유의수준 0.05에서 유의하므로
c_thickness에 유의미한 영향력을 있다고 할 수 있다.

→모든 변수들의 계수가 음수이므로,
공정변수들이 커질 때
c_thickness는 얇아지는 경향이 있음을 알 수 있다.

→ F-통계량 및 유의확률을 확인할 때,
모형 전체도 매우 유의함을 알 수 있다.

M/L: 회귀분석
1.선형회귀모형(Linear Model)

선형회귀모형의 적합 및 변수선택

- 이를 위해 leap패키지에 있는 regsubsets() 함수를 사용한다.
- regsubsets()함수는 best subset selection을 수행하며,
- 모형의 크기(변수가 포함되는 개수) 순으로 가장 좋은 모형을 출력함

```
vs1 <- regsubsets(c_thickness~.,data=car1)
```

```
Subset selection object
Call: regsubsets.formula(c_thickness ~ ., data = car1)
6 Variables  (and intercept)
                 Forced in Forced out
fix_time             FALSE      FALSE
a_speed              FALSE      FALSE
b_speed              FALSE      FALSE
mpa                  FALSE      FALSE
load_time            FALSE      FALSE
highpressure_time    FALSE      FALSE
1 subsets of each size up to 6
Selection Algorithm: exhaustive
         fix_time a_speed b_speed mpa load_time highpressure_time
1  ( 1 ) " "      " "     " "     "*" " "       " "
2  ( 1 ) " "      " "     "*"     "*" " "       " "
3  ( 1 ) " "      " "     "*"     "*" "*"       " "
4  ( 1 ) " "      "*"     "*"     "*" "*"       " "
5  ( 1 ) "*"      "*"     "*"     "*" "*"       " "
6  ( 1 ) "*"      "*"     "*"     "*" "*"       "*"
```

→변수 1개만 모형에 포함하는 경우에는
mpa이 선택되었음으로, 가장 영향력이 큰 공정변수로 볼 수 있다.

→ 두 개의 변수만 포함하는 경우에는
mpa 와 b_speed가 포함되었다.

이는 탕구두께와 개별적인 상관계수를 계산했을 때
'실압력과 a속도가 가장 큰 상관관계를 가짐' 와는
약간 다른 결과이다.

M/L: 회귀분석
1.선형회귀모형(Linear Model)

선형회귀모형의 적합 및 변수선택

- 이를 위해 MASS패키지에 있는 stepAIC()함수를 사용한다.
- 'directions'을 'forward', 'backward', 'both' 3가지로 조정하여 수행한다.

```
vs2 <- stepAIC(lm11,direction="both", scope=list(upper=lm1,lower=lm11))
vs21 <- stepAIC(lm11,direction="forward", scope=list(upper=lm1,lower=lm11))
vs22 <- stepAIC(lm1,direction="backward")
```

```
> vs2 <- stepAIC(lm11,direction="both", scope=list(upper=lm1,lower=lm11))
Start:  AIC=5978.38
c_thickness ~ 1

                    Df Sum of Sq   RSS    AIC
+ mpa                1    9241.8 10342 3789.8
+ a_speed            1    7628.9 11955 4287.0
+ load_time          1    1205.2 18379 5762.5
+ b_speed            1     783.6 18801 5840.3
+ fix_time           1     238.8 19346 5938.3
+ highpressure_time  1      14.6 19570 5977.8
<none>                          19584 5978.4

Step:  AIC=3789.81
c_thickness ~ mpa

                    Df Sum of Sq     RSS    AIC
+ b_speed            1     826.6  9515.9 3506.0
+ load_time          1     430.1  9912.5 3646.1
+ fix_time           1     210.7 10131.8 3721.2
+ a_speed            1      80.5 10262.0 3765.0
+ highpressure_time  1      63.2 10279.4 3770.8
<none>                          10342.5 3789.8
- mpa                1    9241.8 19584.3 5978.4

Step:  AIC=3506.01
c_thickness ~ mpa + b_speed
```

```
> vs21 <- stepAIC(lm11,direction="forward", scope=list(upper=lm1,lower=lm11))
Start:  AIC=5978.38
c_thickness ~ 1

                    Df Sum of Sq   RSS    AIC
+ mpa                1    9241.8 10342 3789.8
+ a_speed            1    7628.9 11955 4287.0
+ load_time          1    1205.2 18379 5762.5
+ b_speed            1     783.6 18801 5840.3
+ fix_time           1     238.8 19346 5938.3
+ highpressure_time  1      14.6 19570 5977.8
<none>                          19584 5978.4

Step:  AIC=3789.81
c_thickness ~ mpa

                    Df Sum of Sq     RSS    AIC
+ b_speed            1     826.61  9515.9 3506.0
+ load_time          1     430.08  9912.5 3646.1
+ fix_time           1     210.71 10131.8 3721.2
+ a_speed            1      80.54 10262.0 3765.0
+ highpressure_time  1      63.17 10279.4 3770.8
<none>                          10342.5 3789.8
```

```
> vs22 <- stepAIC(lm1,direction="backward")
Start:  AIC=3291.81
c_thickness ~ fix_time + a_speed + b_speed + mpa + load_time +
    highpressure_time

                    Df Sum of Sq     RSS    AIC
<none>                          8919.2 3291.8
- highpressure_time  1     48.90  8968.1 3308.6
- fix_time           1     69.92  8989.1 3316.6
- a_speed            1    105.79  9025.0 3330.3
- load_time          1    196.43  9115.6 3364.6
- b_speed            1    790.81  9710.0 3581.3
- mpa                1   1618.55 10537.7 3862.0
```

→3가지 모두 같은 결과를 보여준다.

```
c_thickness ~ mpa + b_speed + load_time + a_speed + fix_time + highpressure_time
```

M/L: 회귀분석

1.선형회귀모형(Linear Model)

다중공선성 파악하기

- 다중공선성이 존재하면,
  추정량의 분산을 크게 하여 모형을 매우 불안정하게 만든다.
- car 패키지에 있는 vif함수를 이용하여
  분산팽창계수를 통해 다중공선성을 파악한다.

```
> vif(lm1)
        fix_time          a_speed          b_speed              mpa        load_time highpressure_time
        1.376796         3.964659         1.012278         3.921777         1.277283         1.014540
```

→ 분산팽창계수가 전반적으로 4를 넘지 않으므로
다중공선성이 심각하게 존재하지 않는 것으로 볼 수 있다.

보통, 분산팽창계수가 10 을 넘으면 심각한 것으로 판단하고
4~5 를 넘으면 의심해 볼 수 있는 것으로 알려져 있다.

M/L: 회귀분석
2. 축소추정법

- 선형회귀모형에서 설명변수의 개수가 다수 존재하거나 변수들간의 상관성이 큰 경우 다중공선성 문제가 발생할 수 있다.
- 축소추정법(shrinkage method)을 통해 이를 해소하므로 모형의 성능을 개선할 있다.

!!! 이미 모든 변수들이 유의미한 것으로 판명되었고,
다중공선성도 크게 의심할 수준이 아니라 필수적인 절차이지는 않지만,
M/L 공부를 위해 해봤습니다.

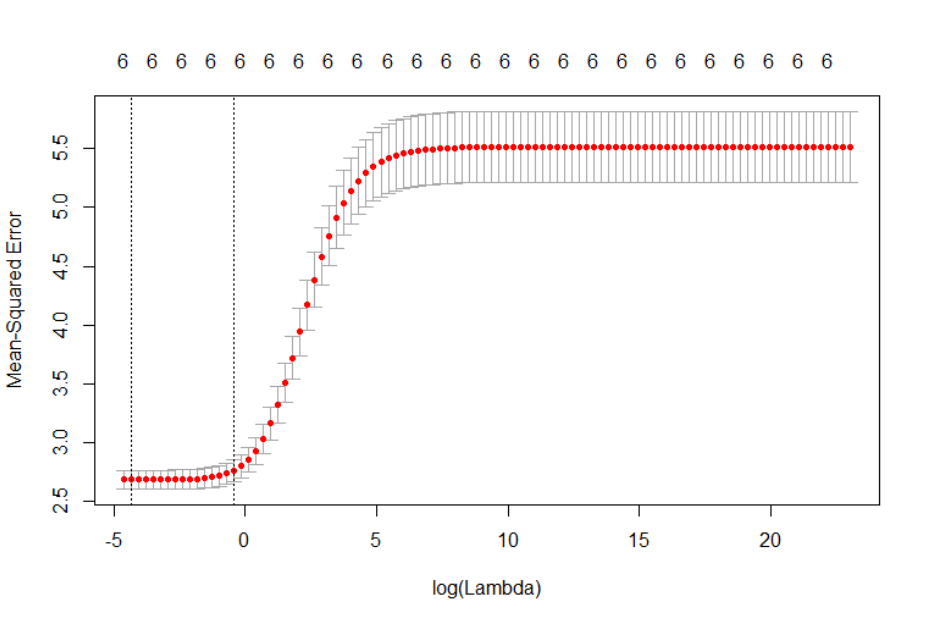
데이터셋 준비

```
library(glmnet)
set.seed(1)
ind.train <- sample(1:nrow(car1),nrow(car1)*0.7)
car.train <- car1[ind.train,]
car.test <- car1[-ind.train,]
X <- as.matrix(car.train[,1:6])
Y <- as.matrix(car.train[,7])
nX <- as.matrix(car.test[,1:6])
nY <- as.matrix(car.test[,7])
```

M/L: 회귀분석
2.1 능형(Ridge)회귀

- 적절한 모수를 찾기 위해 교차검증을 실시한다.
- 최적 모수를 이용하여 능형회귀 모델을 예측한다.

```
cv.ridge <- cv.glmnet(X,Y,alpha=0,lambda=10^seq(10,-2,length=100))
ridge.pred <- predict(glmnet(X,Y,alpha=0,lambda=cv.ridge$lambda.min),newx=nX)
# Ridge 모형적합을 위해 alpha=0 으로 설정
mean((nY - ridge.pred)^2)
coef(glmnet(X,Y,alpha=0,lambda=cv.ridge$lambda.min))
```



→ 그림에서 볼 수 있듯이 조절 모수의 값은 작은 쪽에서 형성된다.

→ 실제 선택된 값은 0.01321941다.

```
$lambda.min
[1] 0.01321941
```

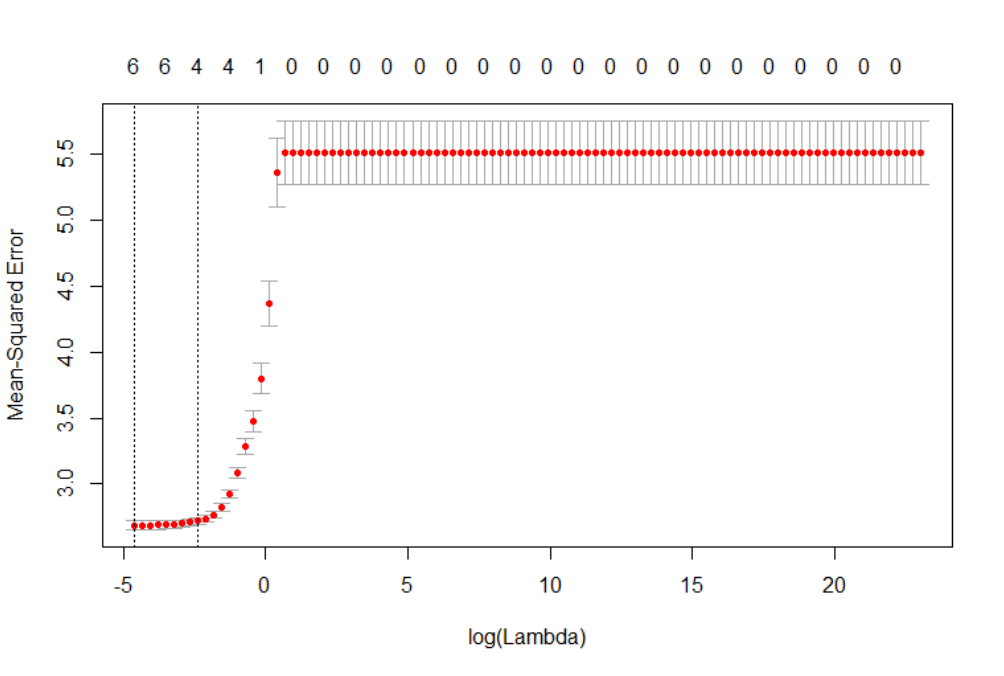
→ 예측오차는 2.4964450이다.

```
➢ mean((nY - ridge.pred)^2)
➢   [1] 2.496445
```

M/L: 회귀분석
2.2 라소(LASSO)회귀

- 적절한 모수를 찾기 위해 교차검증을 실시한다.
- 최적 모수를 이용하여 라소회귀 모델을 예측한다.

```
cv.lasso <- cv.glmnet(X, Y, alpha=1, lambda=10^seq(10, -2, length=100))
lasso.pred <- predict(glmnet(X,Y,alpha=1,lambda=cv.lasso$lambda.min),newx=nX)
mean((nY - lasso.pred)^2)
coef(glmnet(X,Y,alpha=1,lambda=cv.lasso$lambda.min))
```



→ 그림에서 볼 수 있듯이 조절 모수의 값은 작은 쪽에서 형성된다.

→ 실제 선택된 값은 0.01로 현재 설정한 값들 중 최소에 해당하는 값이다.

```
$lambda.min
[1] 0.01
```

→ 추정된 값중 0은 없다. 즉, 모형에서 빠지는 변수는 없다.

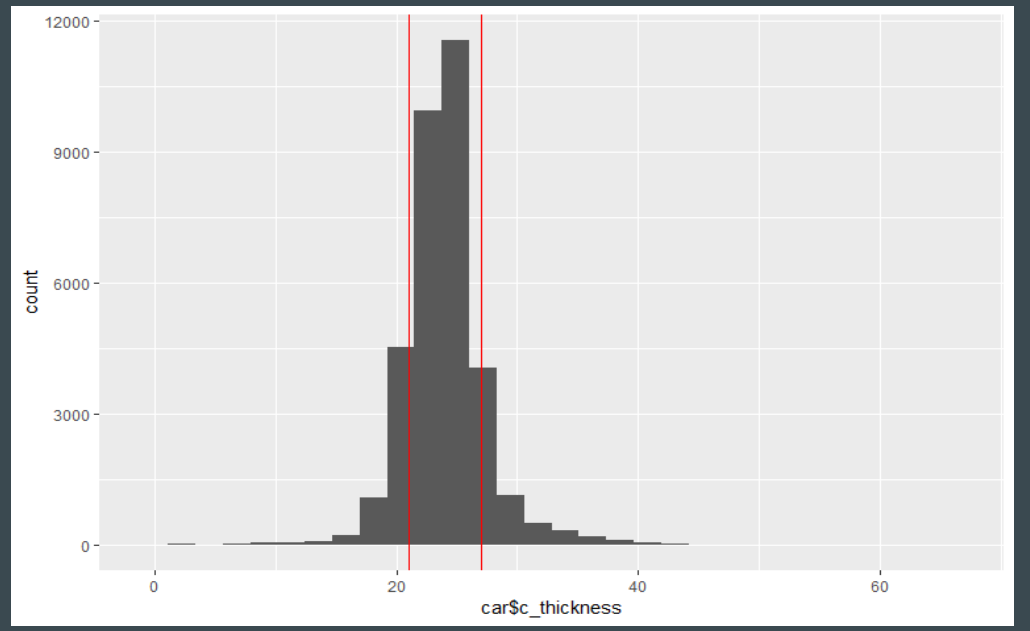
→ 예측오차는 2.500823이다.

```
➢ mean((nY - lasso.pred)^2)
➢ [1] 2.500823
```

M/L: 분류분석

불량 vs 양품 분석

- 탕구두께가 [21,27] 구간을 벗어나는 것은 불량이 의심된다.
- 탕구두께가 [21,27] 구간에 속하는지 여부를
  반응변수(불량품or양품)로
  반응변수을 이산형(binary)으로 변환하여 분류문제로 분석한다.




```
car1$failure <- as.factor((car1$c_thickness > 27)+(car1$c_thickness < 21))
car2 <- car1[,-7]
head(car2)
table(car2$failure)
car2.train <- car2[ind.train,]
car2.test <- car2[-ind.train,]
```

```
> head(car2)
   fix_time a_speed b_speed  mpa load_time highpressure_time failure
8      68.3   0.492   1.702 29.6       0.0                96       1
9      81.4   0.502   1.702 29.8      19.1                96       1
10     81.4   0.651   1.689 26.3      19.1                96       1
14     81.8   0.660   1.722 76.2      20.2                88       1
16     81.6   0.667   1.712 76.6      20.3                87       1
17     81.6   0.672   1.692 76.0      20.2                87       1
> table(car2$failure)

   0    1
3051  380
```

M/L: 분류분석
1. 로지스틱(Logistic) 회귀모형

- 로지스틱 회귀모형은 선형모형의 확장된 형태이며,
  불량발생확률의 logit 변환을 공정변수들의 선형결합으로 모형화한 것이다.
- 선형회귀모형과 비슷한 함수들을 이용하여 적합 및 변수 선택이 가능하다.

```
lm2 <- glm(failure~.,data=car2.train, family=binomial)
```

```
Call:
glm(formula = failure ~ ., family = binomial, data = car2.train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.5986  -0.3920  -0.3315  -0.2837   3.5042

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)       -22.63930    6.00754  -3.768 0.000164 ***
fix_time            0.22062    0.07844   2.813 0.004914 **
a_speed             1.16541    4.88318   0.239 0.811371
b_speed             1.65214    0.24259   6.810 9.73e-12 ***
mpa                -0.10978    0.01861  -5.899 3.66e-09 ***
load_time          -0.12995    0.07221  -1.800 0.071900 .
highpressure_time   0.13252    0.02038   6.503 7.87e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1671.9  on 2400  degrees of freedom
Residual deviance: 1171.0  on 2394  degrees of freedom
AIC: 1185

Number of Fisher Scoring iterations: 6
```

→ 유의수준 0.05에서
a_speed와 road_time 이 유의하지 않은 것으로 나왔다.

M/L: 분류분석
1. 로지스틱(Logistic) 회귀모형

선형회귀모형의 적합 및 변수선택

- 이를 위해 MASS패키지에 있는 stepAIC()함수를 사용한다.
- 'directions'을 'forward', 'backward', 'both' 3가지로 조정하여 수행한다.

```
vs3 <- stepAIC(lm2,direction="both")
vs31 <- stepAIC(lm22, direction = "forward", scope=list(upper=lm2,lower=lm22))
vs33 <- stepAIC(lm2, direction = "backward")
```

```
> vs3 <- stepAIC(lm2,direction="both")
Start:  AIC=1184.95
failure ~ fix_time + a_speed + b_speed + mpa + load_time + highpressure_time

                   Df Deviance    AIC
- a_speed           1   1171.0 1183.0
<none>              1171.0 1185.0
- load_time         1   1174.2 1186.2
- fix_time          1   1182.3 1194.3
- highpressure_time 1   1212.0 1224.0
- b_speed           1   1213.1 1225.1
- mpa               1   1252.6 1264.6

Step:  AIC=1183.01
failure ~ fix_time + b_speed + mpa + load_time + highpressure_time

                   Df Deviance    AIC
<none>              1171.0 1183.0
- load_time         1   1174.3 1184.3
+ a_speed           1   1171.0 1185.0
- fix_time          1   1182.3 1192.3
- highpressure_time 1   1212.1 1222.1
- b_speed           1   1213.1 1223.1
- mpa               1   1459.3 1469.3
```

```
> vs31 <- stepAIC(lm22, direction = "forward", scope=list(upper=lm2,lower=lm22))
Start:  AIC=1673.85
failure ~ 1

                   Df Deviance    AIC
+ mpa               1   1283.4 1287.4
+ a_speed           1   1376.0 1380.0
+ fix_time          1   1594.4 1598.4
+ highpressure_time 1   1612.3 1616.3
+ b_speed           1   1623.1 1627.1
+ load_time         1   1663.4 1667.4
<none>              1671.8 1673.8

Step:  AIC=1287.38
failure ~ mpa

                   Df Deviance    AIC
+ highpressure_time 1   1222.4 1228.4
+ b_speed           1   1228.8 1234.8
+ fix_time          1   1274.9 1280.9
<none>              1283.4 1287.4
+ load_time         1   1283.0 1289.0
+ a_speed           1   1283.4 1289.4
```

```
> vs33 <- stepAIC(lm2, direction = "backward")
Start:  AIC=1184.95
failure ~ fix_time + a_speed + b_speed + mpa + load_time + highpressure_time

                   Df Deviance    AIC
- a_speed           1   1171.0 1183.0
<none>              1171.0 1185.0
- load_time         1   1174.2 1186.2
- fix_time          1   1182.3 1194.3
- highpressure_time 1   1212.0 1224.0
- b_speed           1   1213.1 1225.1
- mpa               1   1252.6 1264.6

Step:  AIC=1183.01
failure ~ fix_time + b_speed + mpa + load_time + highpressure_time

                   Df Deviance    AIC
<none>              1171.0 1183.0
- load_time         1   1174.3 1184.3
- fix_time          1   1182.3 1192.3
- highpressure_time 1   1212.1 1222.1
- b_speed           1   1213.1 1223.1
- mpa               1   1459.3 1469.3
```

→ AIC에 의한 변수선택결과 a_speed를 제외한 모형이 최적으로 나왔다.

`failure ~ fix_time + b_speed + mpa + load_time + highpressure_time`

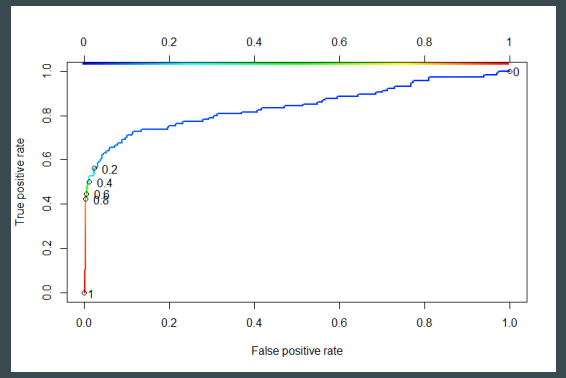
M/L: 분류분석
1. 로지스틱(Logistic) 회귀모형

모델 평가하기

- test Data로 예측하기
- ROC Curve
- AUC
- Confusion Matrix

```
fit.log <- predict.glm(vs3, newdata=car2.test, type="response")
#roc
pred.log <- prediction(fit.log, car2.test$failure)
perf.log <- performance(pred.log, "tpr", "fpr")
plot(perf.log, colorize=T, colorkey.pos="top", print.cutoffs.at=seq(0, 1, by=0.2),
     text.cex=1,text.adj=c(-0.5, 0.5), lwd=2)
#auc
auc <- performance(pred.log,"auc")
auc
#confusionMatrix
a<-as.factor(1*(fit.log>0.5))
confusionMatrix(a, car2.test$failure)
```

1. ROC Curve_Logistic



2. AUC

```
Slot "y.values":
 [[1]] [1] 0.8383322
```

3. Confusion Matrix

```
> confusionMatrix(a, car2.test$failure)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 910  61
         1   6  53

               Accuracy : 0.935
                 95% CI : (0.9181, 0.9492)
    No Information Rate : 0.8893
    P-Value [Acc > NIR] : 3.852e-07

                  Kappa : 0.5811
 Mcnemar's Test P-Value : 4.191e-11

            Sensitivity : 0.9934
            Specificity : 0.4649
         Pos Pred Value : 0.9372
         Neg Pred Value : 0.8983
             Prevalence : 0.8893
         Detection Rate : 0.8835
   Detection Prevalence : 0.9427
      Balanced Accuracy : 0.7292

       'Positive' Class : 0
```

→ AUC값은 0.8383322이다.

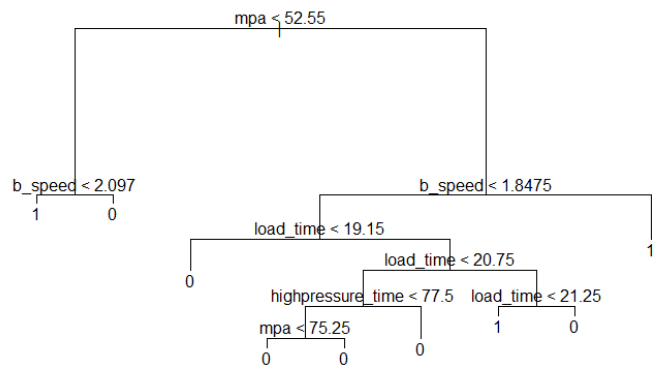
→로지스틱 모형에 의한
오분류율 (1-0.935) 은 0.065

→ 양품은 양품으로 대부분 잘 분류하였으나 (sensitivity=0.9934),
불량품에 대한 분류성능이 떨어지는 것으로 보인다 (specificity=0.4649).

M/L: 분류분석
2. 의사결정나무 (Decision tree)

- 로지스틱 모형과 달리 비선형모형이다.
- 공정변수들의 값이 이원화되어 있는 경우가 많았으므로
- 로지스틱 모형보다는 의사결정나무 모형에 의한 분류성능이 더 좋을 것으로 기대해 볼 수 있다.

```
tree.car <- tree(failure~., data=car2.train)
summary(tree.car)

Classification tree:
tree(formula = failure ~ ., data = car2.train)
Variables actually used in tree construction:
[1] "mpa"                "b_speed"            "load_time"          "highpressure_time"
Number of terminal nodes:  9
Residual mean deviance:  0.3672 = 878.4 / 2392
Misclassification error rate: 0.06456 = 155 / 2401
```



→ 의사결정나무 결과, 실제 나무적합을 위해 쓰인 변수는
mpa, b_speed, highpressure_time, road_tie, fix_time이다.

→ a속도는 사용되지 않았다.

→ 보통 큰 나무모형은 과적합(overfitting) 문제가 있는 것으로 알려져 있다.

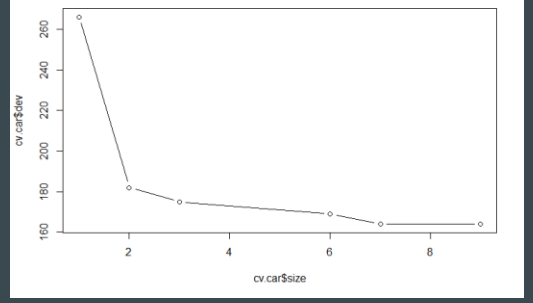
→ 적절한 가지치기(pruning)를 통해 나무의 크기를 줄여 예측 정확도
(1-0.06456=0.93544)를 향상시켜 볼 것이다.

M/L: 분류분석
2. 1.가지치기

의사결정트리 가지치기

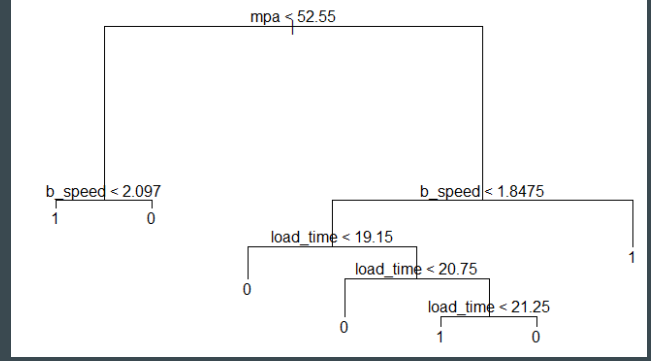
- 적절한 가지의 수를 찾기 위해 교차검증을 실시한다.
- 최적의 모수로 가지치기를 한다.

```
cv.car <- cv.tree(tree.car, FUN=prune.misclass)
> cv.car
$`size`
[1] 9 7 6 3 2 1

$dev
[1] 164 164 169 175 182 266
```



→ 크기가 7인 경우 deviance가 최소가 된다

```
prune.car <- prune.misclass(tree.car, best=7)
>prune.car
node), split, n, deviance, yval, (yprob)
      * denotes terminal node

 1) root 2401 1672.000 0 ( 0.88921 0.11079 )
   2) mpa < 52.55 102   56.080 1 ( 0.07843 0.92157 )
     4) b_speed < 2.097 97   33.340 1 ( 0.04124 0.95876 ) *
     5) b_speed > 2.097 5    5.004 0 ( 0.80000 0.20000 ) *
   3) mpa > 52.55 2299 1223.000 0 ( 0.92518 0.07482 )
     6) b_speed < 1.8475 2246 1048.000 0 ( 0.93767 0.06233 )
      12) load_time < 19.15 98  125.200 0 ( 0.66327 0.33673 ) *
      13) load_time > 19.15 2148  850.500 0 ( 0.95019 0.04981 )
        26) load_time < 20.75 2079  671.700 0 ( 0.96200 0.03800 ) *
        27) load_time > 20.75 69   93.190 0 ( 0.59420 0.40580 )
          54) load_time < 21.25 39   50.920 1 ( 0.35897 0.64103 ) *
          55) load_time > 21.25 30   19.500 0 ( 0.90000 0.10000 ) *
     7) b_speed > 1.8475 53   71.170 1 ( 0.39623 0.60377 ) *
```



크기가 7인 모형을 선택한다.

분류를 위해 쓰인 변수는
mpa, road_time, b_speed 세 개이다

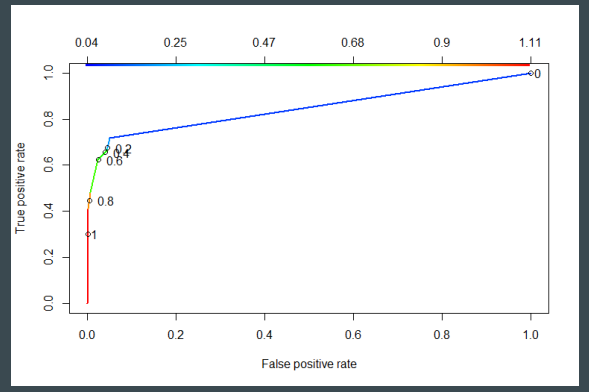
M/L: 분류분석
2. 의사결정나무 (Desicion Tree)

모델 평가하기

- test Data로 예측하기
- ROC Curve
- AUC
- Confusion Matrix

```
fit.tree <- predict(prune.car, newdata=car2.test, type="vector")
#roc
pred.tree <- prediction(fit.tree[,2], car2.test$failure)
perf.tree <- performance(pred.tree, "tpr", "fpr")
plot(perf.tree, colorize=T, colorkey.pos="top", print.cutoffs.at=seq(0,1,by=0.2),
     text.cex=1, text.adj=c(-0.5, 0.5), lwd=2)
#auc
perf.tree1 <- performance(pred.tree, "auc")
perf.tree1
#confusionMatrix
confusionMatrix(as.factor(1*(fit.tree[,2]>0.5)), car2.test$failure)
```

1. ROC Curve_Logistic



2. AUC

```
Slot "y.values":
[[1]]
[1] 0.8456868
```

3. Confusion Matrix

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 894  43
         1  22  71

               Accuracy : 0.9369
                 95% CI : (0.9203, 0.951)
    No Information Rate : 0.8893
    P-Value [Acc > NIR] : 1.139e-07

                  Kappa : 0.6513
 Mcnemar's Test P-Value : 0.01311

            Sensitivity : 0.9760
            Specificity : 0.6228
         Pos Pred Value : 0.9541
         Neg Pred Value : 0.7634
             Prevalence : 0.8893
         Detection Rate : 0.8680
   Detection Prevalence : 0.9097
      Balanced Accuracy : 0.7994

       'Positive' Class : 0
```

→ AUC값은 0.8456868이다.

→로지스틱 모형에 의한
오분류율 (1-0.936) 은 0.064

→ 양품은 양품으로 대부분 잘 분류하였으나 (sensitivity=0.9760),
불량품에 대한 분류성능이 떨어지는 것으로 보인다 (specificity=0.6228).

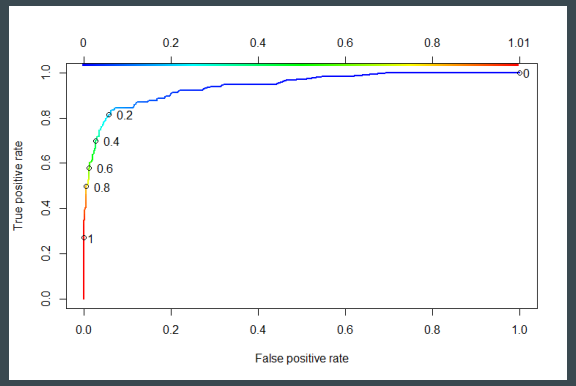
M/L: 분류분석
3. 랜덤포레스트 (Random forest)

모델 평가하기

- test Data로 예측하기
- ROC Curve
- AUC
- Confusion Matrix

```
fit.rf <- predict(rf.car,newdata=car2.test,type="prob")
#roc
pred.rf <- prediction(fit.rf[,2], car2.test$failure)
perf.rf <- performance(pred.rf,"tpr","fpr")
plot(perf.rf, colorize=T, colorkey.pos="top", print.cutoffs.at=seq(0,1,by=0.2),
     text.cex=1,text.adj=c(-0.5, 0.5), lwd=2)
#auc
perf.rf1 <- performance(pred.rf,"auc")
perf.rf1
#confusionMatrix
confusionMatrix(as.factor(1*(fit.rf[,2]>0.5)),car2.test$failure)
```

1. ROC Curve_Logistic



2. AUC

```
Slot "y.values":
[[1]]
[1] 0.9410432
```

3. Confusion Matrix

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 898  44
         1  18  70

               Accuracy : 0.9398
                 95% CI : (0.9235, 0.9535)
    No Information Rate : 0.8893
    P-Value [Acc > NIR] : 1.624e-08

                  Kappa : 0.6603
 Mcnemar's Test P-Value : 0.001498

            Sensitivity : 0.9803
            Specificity : 0.6140
         Pos Pred Value : 0.9533
         Neg Pred Value : 0.7955
             Prevalence : 0.8893
         Detection Rate : 0.8718
   Detection Prevalence : 0.9146
      Balanced Accuracy : 0.7972

       'Positive' Class : 0
```

→ AUC값은 0.9410432이다.

→로지스틱 모형에 의한
오분류율 (1-0.939) 은 0.061

→ 양품은 양품으로 대부분 잘 분류하였으나 (sensitivity=0.9803),
불량품에 대한 분류성능이 떨어지는 것으로 보인다 (specificity=0.6140).

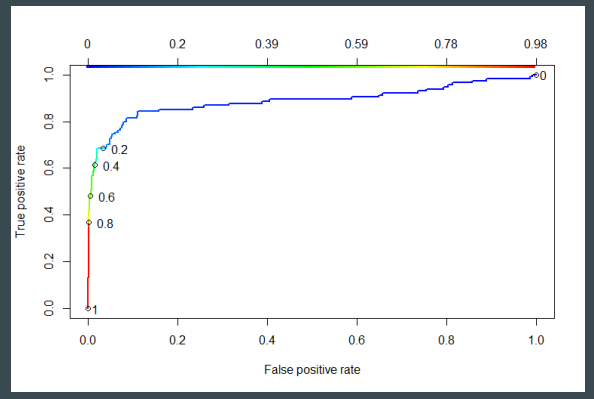
M/L: 분류분석
4. 신경망모형 (Neural network model)

모델 평가하기

- test Data로 예측하기
- ROC Curve
- AUC
- Confusion Matrix

```
#roc
pred.nn <- prediction(fit.nn[,2], car2.test$failure)
perf.nn <- performance(pred.nn,"tpr","fpr")
plot(perf.nn, colorize=T, colorkey.pos="top", print.cutoffs.at=seq(0,1,by=0.2),
     text.cex=1,text.adj=c(-0.5, 0.5), lwd=2)
#auc
perf.nn1 <- performance(pred.nn,"auc")
perf.nn1
#confusionMatrix
caret::confusionMatrix(as.factor(1*(fit.nn[,2]>0.5)) ,car2.test$failure)
```

1. ROC Curve_Logistic



2. AUC

```
Slot "y.values":
[[1]]
[1] 0.8867023
```

3. Confusion Matrix

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 905  48
         1  11  66

               Accuracy : 0.9427
                 95% CI : (0.9267, 0.9561)
    No Information Rate : 0.8893
    P-Value [Acc > NIR] : 1.992e-09

                  Kappa : 0.6608
 Mcnemar's Test P-Value : 2.775e-06

            Sensitivity : 0.9880
            Specificity : 0.5789
         Pos Pred Value : 0.9496
         Neg Pred Value : 0.8571
             Prevalence : 0.8893
         Detection Rate : 0.8786
   Detection Prevalence : 0.9252
      Balanced Accuracy : 0.7835

       'Positive' Class : 0
```

→ AUC값은 0.8867023이다.

→로지스틱 모형에 의한
오분류율 (1-0.942) 은 0.058

→ 양품은 양품으로 대부분 잘 분류하였으나 (sensitivity=0.9880),
불량품에 대한 분류성능이 떨어지는 것으로 보인다 (specificity=0.5789).

Estimation: 회귀분석모델 간의 오차 비교

선형회귀분석	Ridge분석	Lasso분석
➢ lm.pred <- predict(lm(c_thickness~.,data=car.train),newdata=car.test[,1:6]) ➢ mean((car.test[,7] -lm.pred)^2) ➢ [1] 2.494666	➢ ridge.pred <- predict(glmnet(X,Y,alpha=0,lambda=cv.ridge$lambda.min),newx=nX) ➢ > mean((nY - ridge.pred)^2) ➢ [1] 2.496445	➢ lasso.pred <- predict(glmnet(X,Y,alpha=1,lambda=cv.lasso$lambda.min),newx=nX) ➢ mean((nY - lasso.pred)^2) ➢ [1] 2.500823

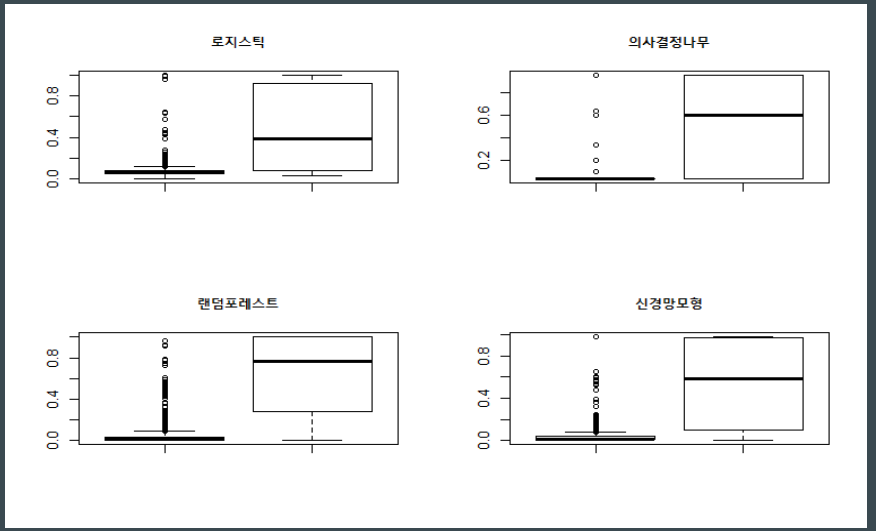
→ 축소추정법을 적용하지 않은 선형회귀모형의 경우 예측오차가 2.4946660이다.
즉, Ridge(2.496445)나 LASSO(2.500823)에 비해 작은 값을 주었다.

→ 앞에서 모두 변수가 유의하였던 점. 축소추정에 의해 변수가 모형에서 빠지지 않았던 점.
축소추정에 의해서 예측오차가 좋아지지 않았던 점. 분산팽창계수가 그리 크지 않았던 점 등을 종합하여 볼 때,
본 자료에서는 축소추정법이 필요하지 않은 것으로 보인다.

→다만, 앞에서 보았듯이 선형모형 가정 자체에도 의심할 만한 여지가 있으며,
공정변수들이 두 그룹으로 완전히 분리되어 나타나는 경우가 많았던 점 등을 생각할 때
선형회귀 모형도 그 성능이 매우 뛰어나다고 보기는 어렵다.

Estimation:분류분석모델 간의 오차 비교

	로지스틱 (Logistic)	의사결정나무 (Decision tree)	랜덤포레스트 (Random forest)	신경망모형 (Neural network model)
AUC	0.8383322	0.8456868	0.9410432	0.8867023
오분류율	0.065	0.064	0.058	0.058
sensitivity	0.9934	0.9760	0.9803	0.9880
specificity	0.4649	0.6228	0.6140	0.5789



→ 위 그림은 양품(좌), 불량의심품(우) 그룹에서 각 모형에 의한 예측확률을 상자그림으로 표현한 것이다.
좌측 상자는 0에 가깝고 우측상자는 1에 가까울 수록,
또한 두 상자의 폭이 좁고 서로 멀리 떨어져 있을 수록 분류가 잘 된 것으로 볼 수 있다.

→ 약간씩의 차이가 있기는 하지만 양품은 불량확률을 0에 매우 가깝게 대부분 예측하고 있음을 알 수 있다.
다만, 양품을 양품으로 가장 잘 예측하는 것은 의사결정나무모형인 것으로 보인다.

→불량의심품의 경우에는 모든 분류모형이 저하된 성능을 보였다.
상대적으로는 로지스틱이 가장 좋지 않은 성능을 보이고 있으며,
의사결정나무 가 가장 좋은 성능을 보여주고 있다.