Multivariate Data Analysis

Assignment #2

고려대학교 공과대학

산업경영공학부

2017170857 이우준

[Q1] 본인이 스스로 Logistic Regression 모델을 적용하는데 적합한 데이터셋을 선정하고 선정이유를 설명하시오.

[A] Breast Cancer Wisconsin (Diagnostic) Data Set을 이번 Logistic Regression 모델을 적용하는데 적합한 데이터셋을 선정하였습니다. Breast Cancer Wisconsin Data Set는 종속변수가 범주형 변수인 (Malignant, Benign)로 이루어져 있어 범주형 변수를 예측하는 Logistic Regression기법을 적용하기에 적합하다 생각하였습니다. 입력변수 또한 30가지 항목으로 이루어진 데이터가 약 600개가 존재하여 모델을 학습시키기 충분하다 판단하여 해당 데이터셋을 선정하였습니다.

[Q2] 해당 데이터셋의 종속변수와 설명변수는 어떤 것들이 있는가? 분석전에 아래 세가지 질문에 대해서 스스로 생각해보고 답변을 하시오.

1. 데이터에서 제공된 설명변수들 중에서 높은 상관관계가 있을 것으로 예상되는 변수들은 어떤 것들이 있는가? 왜 그렇게 생각하는가?

2. 제공된 설명변수들 중에서 종속변수를 예측하는데 필요하지 않을 것으로 예상되는 변수들은 어떤 것들이 있는가? 왜 그렇게 생각하는가?

[A] 해당 데이터셋의 설명변수에는 "radius" ,"texture", "perimeter", "area", "smoothness", "compactness", "concavity", "concave points", "symmetry", "fractal dimension" 이들의 항목의 평균값 (10가지), 표준오차(10가지), 극값(10가지) 이렇게 총 세가지의 항목으로 총 30가지의 설명변수가 존재한다. 종속변수는 Diagnosis 값 즉 진단 결과 (Malignant or Benign) 입니다.

유방암을 대표적인 증세에는 "유방에 멍울이 만져진다, 한쪽 유방의 크기가 평소보다 커지거나 늘어졌다" 가 있습니다. 이와 연관되어 봤을 때 멍울이 만져진다면 texture와 concavity, concave points, smoothness, compactness의 값과의 상관관계가, 한쪽 유방의 크기가 평소보다 커지거나 늘어진다면, symmetry 값이 상관관계가 있을 것이라 예상됩니다.

이를 제외한 radius, perimeter, area, concavity, concave points, fractal dimension 은 유방암의 대표적인 증세를 설명해주지 못해 종속변수를 예측하는데 필요하지 않을 것이라고 예상됩니다.

[Q3] 모든 연속형 숫자 형태를 갖는 (명목형 변수 제외) 개별 입력변수들에 대하여 각각 다음과 같은 단변량 통계량을 계산하고 Box plot을 도시하시오: Mean, Standard deviation, Skewness, Kurtosis. 전체 변수 중에서 정규 분포를 따른다고 할 수 있는 변수들은 몇 개인가? 정규 분포를 따른다고 가정한 근거는 무엇인가?

[A]

```
#Q3&4
mtable<-numeric(30)
stdtable<-numeric(30)
skwtable<-numeric(30)
kurtable<-numeric(30)
rownames(shapiro_test) <- names(bc_input)
colnames(shapiro_test) <- c("W", "p-val")
outliers <- matrix(0, nrow = 30, ncol = 2)
rownames(outliers) <- names(bc_input)
colnames(outliers) <- c("LCL", "UCL")
for(i in 1:30){
  mtable[i]<-mean(bc_input[,i])
  stdtable[i]<-sd(bc_input[,i])
  skwtable[i]<-skewness(bc_input[,i])
  kurtable[i]<-kurtosis(bc_input[,i])
  boxplot(main=names(bc_input[i]), bc_input[,i])
  outliers[i,]<-boxplot_outliers(bc_input[,i])
}
q3table<-data.frame(mtable,stdtable,skwtable,kurtable)
dimnames(q3table)=list(row=colnames(bc_input),col=c("mean","std","skw","kurt"))
```

-코드-

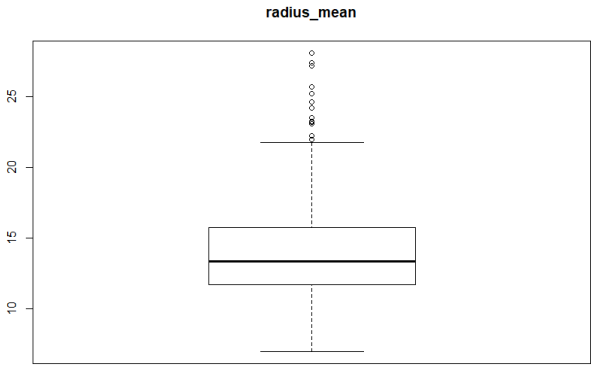
해당 코드를 통해 각 입력변수의 Mean, Standard deviation, Skewness, Kurtosis의 값을 구하면 아래의 표와 같은 결과가 나옵니다.

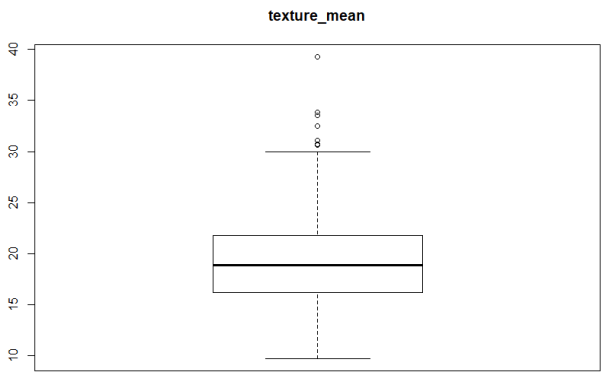
	mean	std	skw	kurt
radius_mean	-1.38E-16	1	0.937417	0.814142
texture_mean	6.16E-17	1	0.647024	0.728007
perimeter_mean	-1.19E-16	1	0.985433	0.939282
area_mean	1.22E-16	1	1.637065	3.586549
smoothness_mean	1.62E-16	1	0.453921	0.824467
compactness_mean	-7.61E-17	1	1.183856	1.608897
concavity_mean	3.93E-17	1	1.393801	1.953136
concave.points_mean	-5.34E-17	1	1.165012	1.032469
symmetry_mean	1.68E-16	1	0.721788	1.251135
fractal_dimension_mean	4.81E-16	1	1.297619	2.948055
radius_se	3.72E-17	1	3.072347	17.44909

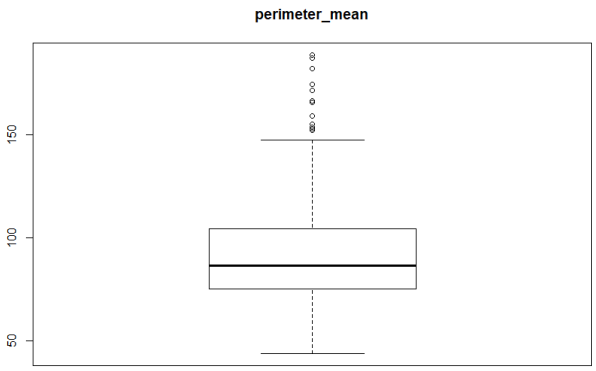
texture_se	-1.07E-16	1	1.637773	5.262633
perimeter_se	1.07E-16	1	3.42548	21.11877
area_se	2.87E-17	1	5.4185	48.5854
smoothness_se	1.29E-16	1	2.302262	10.32059
compactness_se	-1.61E-17	1	1.892203	5.022692
concavity_se	-5.98E-17	1	5.08355	48.24197
concave.points_se	7.92E-18	1	1.43707	5.042496
symmetry_se	9.13E-17	1	2.183573	7.778402
fractal_dimension_se	1.56E-18	1	3.903304	25.93797
radius_worst	-8.48E-17	1	1.097306	0.911503
texture_worst	9.04E-18	1	0.495697	0.20053
perimeter_worst	5.66E-17	1	1.122223	1.036019
area_worst	6.73E-18	1	1.849581	4.321528
smoothness_worst	-2.27E-16	1	0.413238	0.490459
compactness_worst	1.26E-17	1	1.465795	2.981042
concavity_worst	9.17E-17	1	1.144179	1.574447
concave.points_worst	1.17E-17	1	0.490021	-0.55
symmetry_worst	2.76E-16	1	1.426376	4.369103
fractal_dimension_worst	2.00E-16	1	1.653824	5.159356

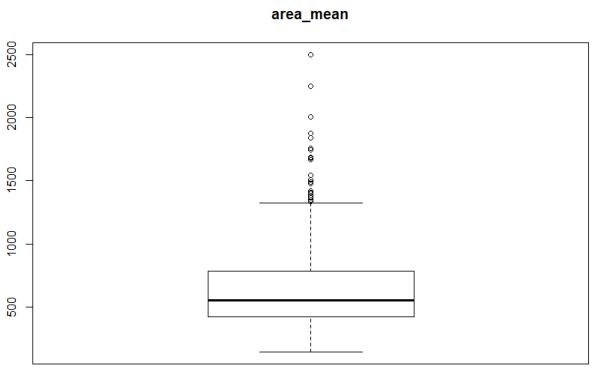
4가지의 지표 중, skewness 값이 +2, -2 사이에 존재하면 치우침이 없는 정규성을 띈 데이터라고 할 수 있습니다. 이 기준을 통해 30가지의 입력변수들 중 정규분포를 따르는 변수는 radius_se, perimeter_se, area_se, smoothness_se, concavity_se, symmetry se 이 6가지의 변수를 제외한 24가지의 변수입니다.

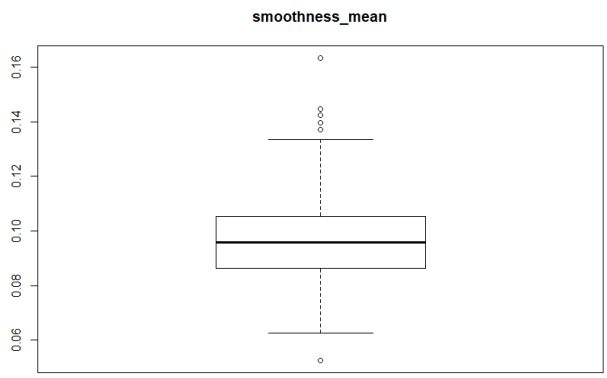
설명변수의 박스플랏은 다음을 통해 확인할 수 있습니다.

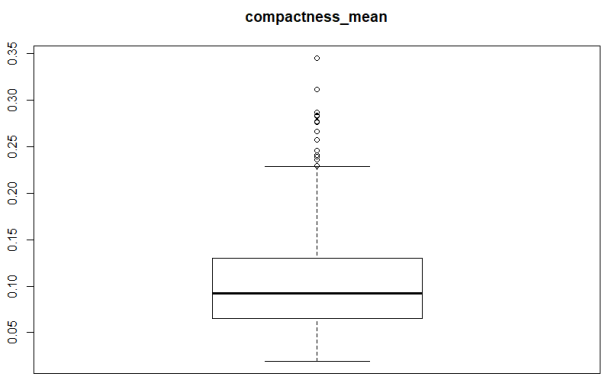


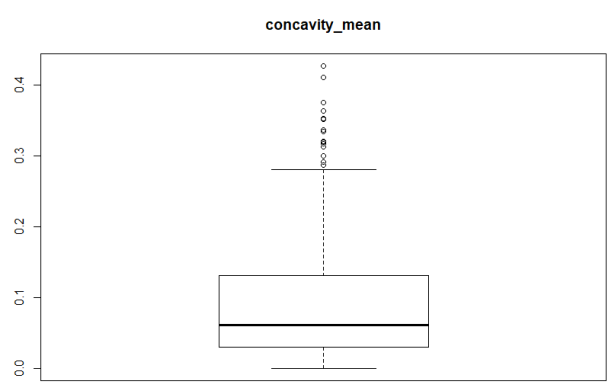


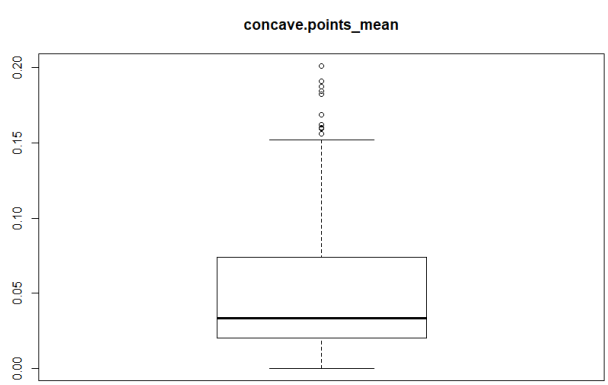

perimeter_mean
50
100
150

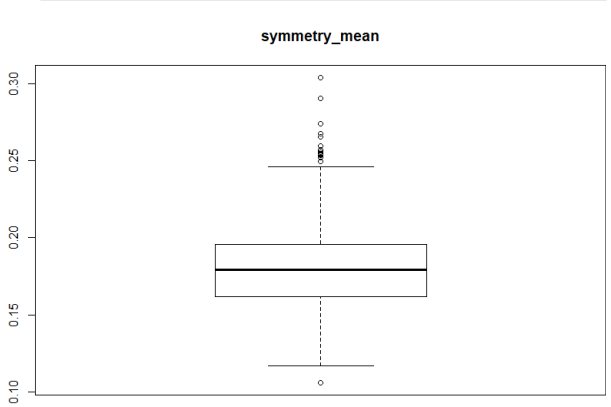

area_mean
500
1000
1500
2000
2500

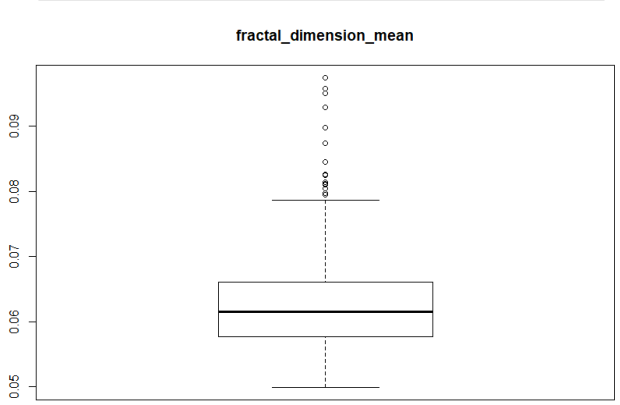

smoothness_mean
0.06
0.08
0.10
0.12
0.14
0.16

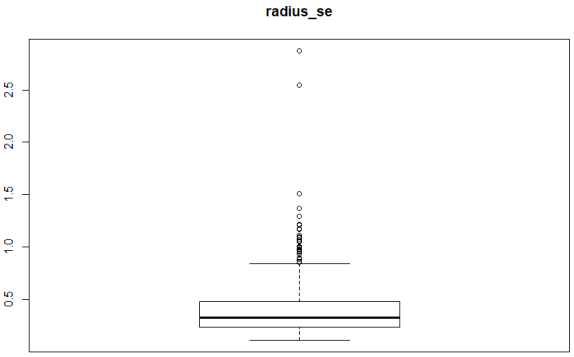

compactness_mean
0.05
0.10
0.15
0.20
0.25
0.30
0.35

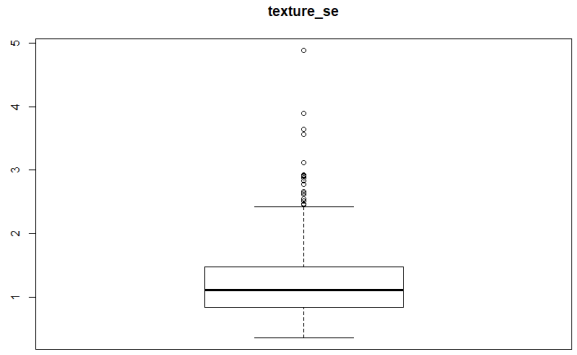

concavity_mean
0.0
0.1
0.2
0.3
0.4

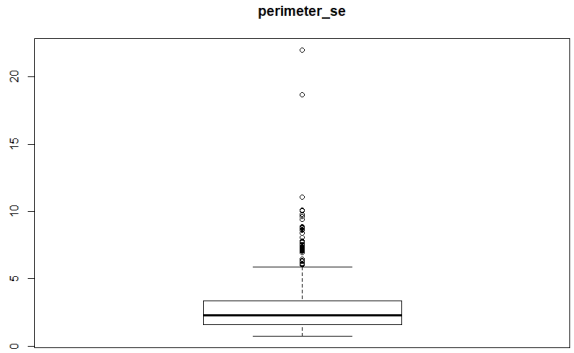

concave.points_mean
0.00
0.05
0.10
0.15
0.20

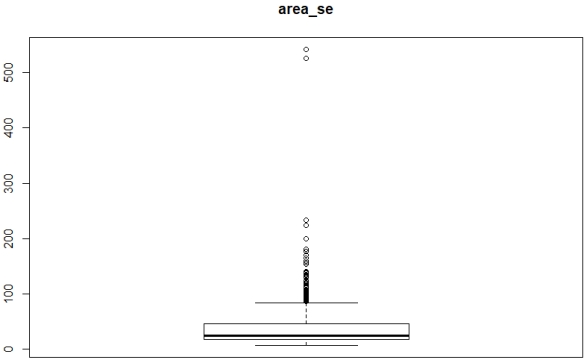

symmetry_mean
0.10
0.15
0.20
0.25
0.30

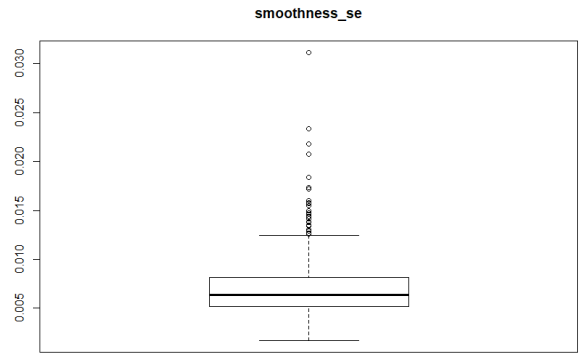

fractal_dimension_mean
0.05
0.06
0.07
0.08
0.09

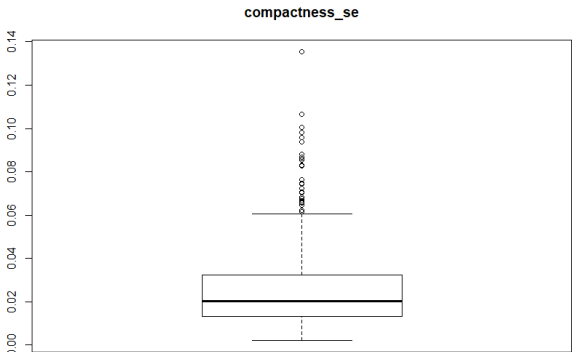

radius_se
0.5
1.0
1.5
2.0
2.5

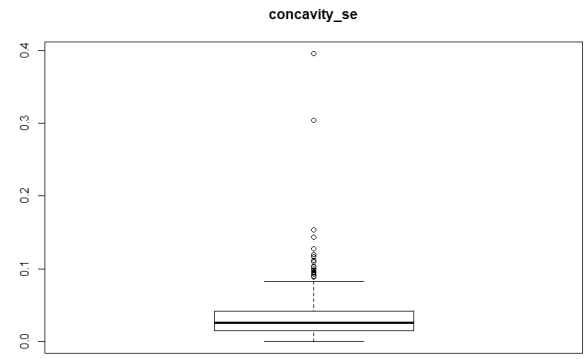

texture_se
1
2
3
4
5

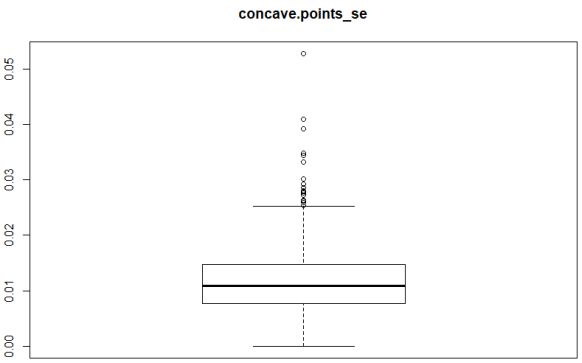

perimeter_se
0
5
10
15
20


area_se
0
100
200
300
400
500

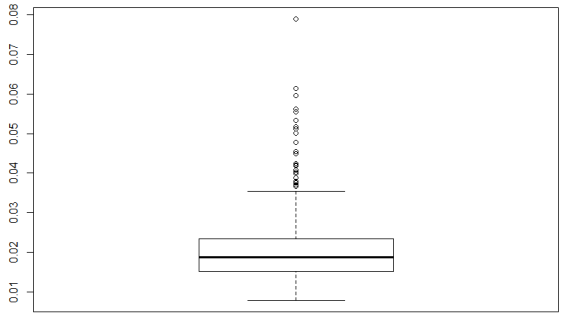

smoothness_se
0.005
0.010
0.015
0.020
0.025
0.030


compactness_se
0.02
0.04
0.06
0.08
0.10
0.12
0.14

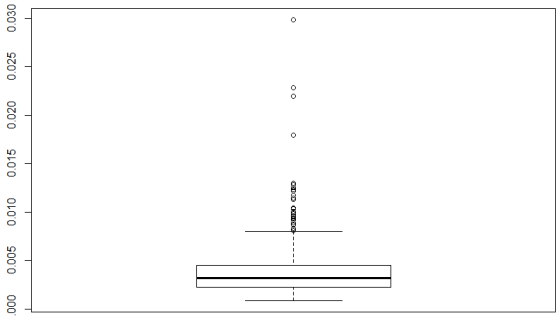

concavity_se
0.0
0.1
0.2
0.3
0.4


concave.points_se
0.00
0.01
0.02
0.03
0.04
0.05

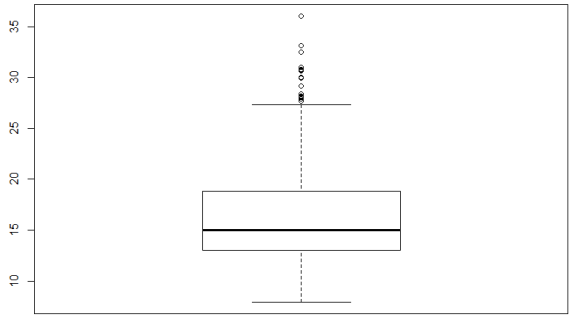
symmetry_se



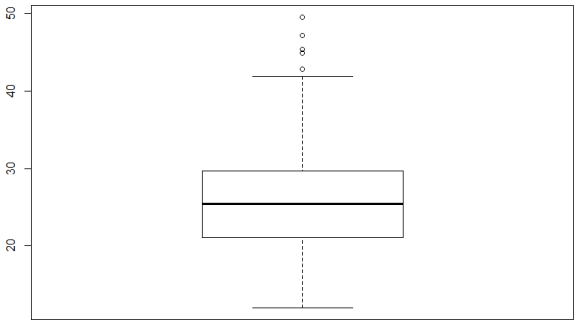
fractal_dimension_se



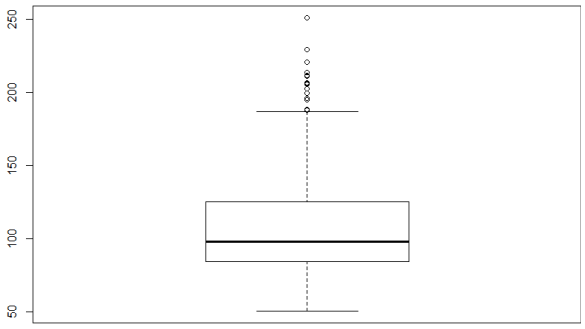
radius_worst



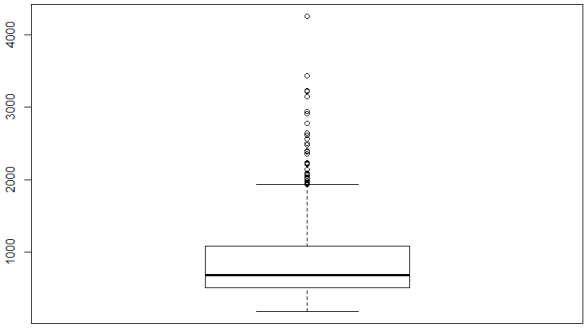
texture_worst



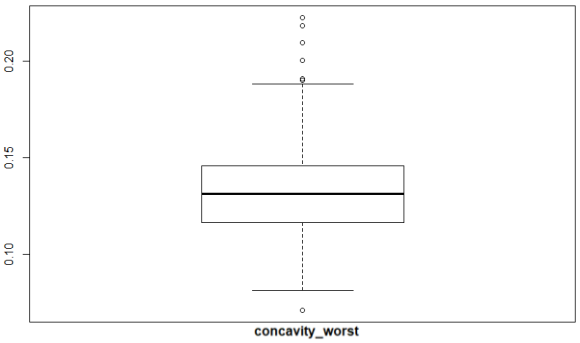
perimeter_worst



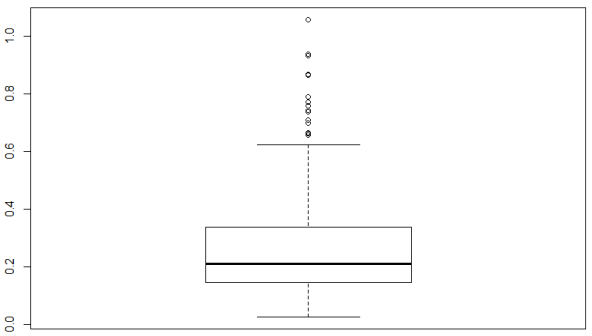
area_worst



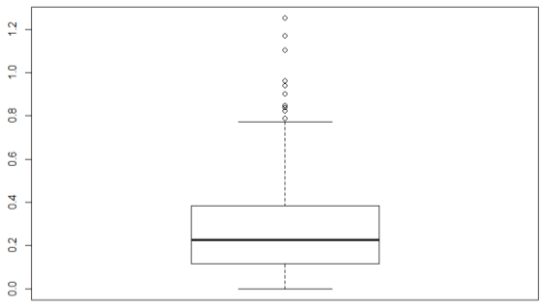
smoothness_worst



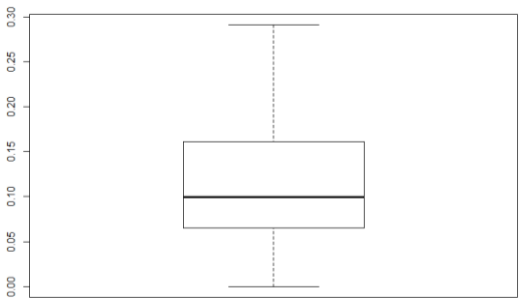
compactness_worst

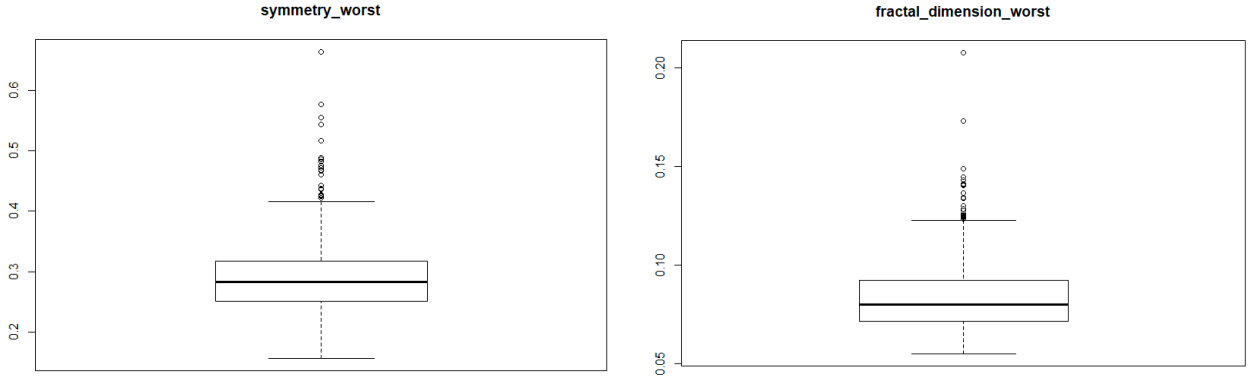


concavity_worst



concave.points_worst





[Q4] [Q3]의Box plot을 근거로 각 변수들에 대한 이상치 (너무 값이 크거나 작은 객체) 조건을 정의하고, 해당하는 객체들을 데이터셋에서 제거해보시오.

[A] 이상치는 각변수의 BoxPlot을 기준으로 볼 때 Whisker 바깥에 존재하는 값으로 이상치 판별의 조건이 되는 Whisker의 값을 표로 정리해보았습니다.

	LCL	UCL
radius_mean	-2.02786	2.163054
texture_mean	-2.22729	2.483205
perimeter_mean	-1.98276	2.27709
area_mean	-1.45316	1.90703
smoothness_mean	-2.40685	2.640741
compactness_mean	-1.60872	2.349035
concavity_mean	-1.11389	2.410953
concave.points_mean	-1.26071	2.656528
symmetry_mean	-2.3514	2.361475
fractal_dimension_mean	-1.81826	2.253764
radius_se	-1.05899	1.577381
texture_se	-1.5529	2.191879
perimeter_se	-1.04313	1.483262
area_se	-0.73718	0.948823
smoothness_se	-1.7745	1.794834
compactness_se	-1.29696	1.962894
concavity_se	-1.05657	1.670516
concave.points_se	-1.91177	2.183669
symmetry_se	-1.53154	1.804625
fractal_dimension_se	-1.096	1.594854

radius_worst	-1.72538	2.286418
texture_worst	-2.22204	2.631321
perimeter_worst	-1.69187	2.367047
area_worst	-1.22135	1.848431
smoothness_worst	-2.23886	2.449648
compactness_worst	-1.44261	2.354412
concavity_worst	-1.30468	2.399105
concave.points_worst	-1.74353	2.683516
symmetry_worst	-2.15906	2.025692
fractal_dimension_worst	-1.60043	2.129097

이 whisker의 값 바깥에 존재해주는 값들을 제거해 보았습니다.

```
for (i in 1:30){
  bc_input[,i]<-ifelse(bc_input[,i] < outliers[i,1] | bc_input[,i] > outliers[i,2], NA,
bc_input[,i])
}
sum(is.na(bc_input))
bc_input_cleared<-na.omit(bc_input)
```

그 결과 총 569개의 데이터에서 398개로 171개의 이상치 데이터가 줄었습니다.

[Q5] 가능한 모든 두 쌍의 입력변수 조합에 대한 산점도(scatter plot) 및 Correlation plot (hint: "corrplot" 패키지의 corrplot( ) 함수사용) 상관관계를 계산해보시오.

1. 어떤 두 조합의 변수들이 서로 강한 상관관계가 있다고 할 수 있는가?

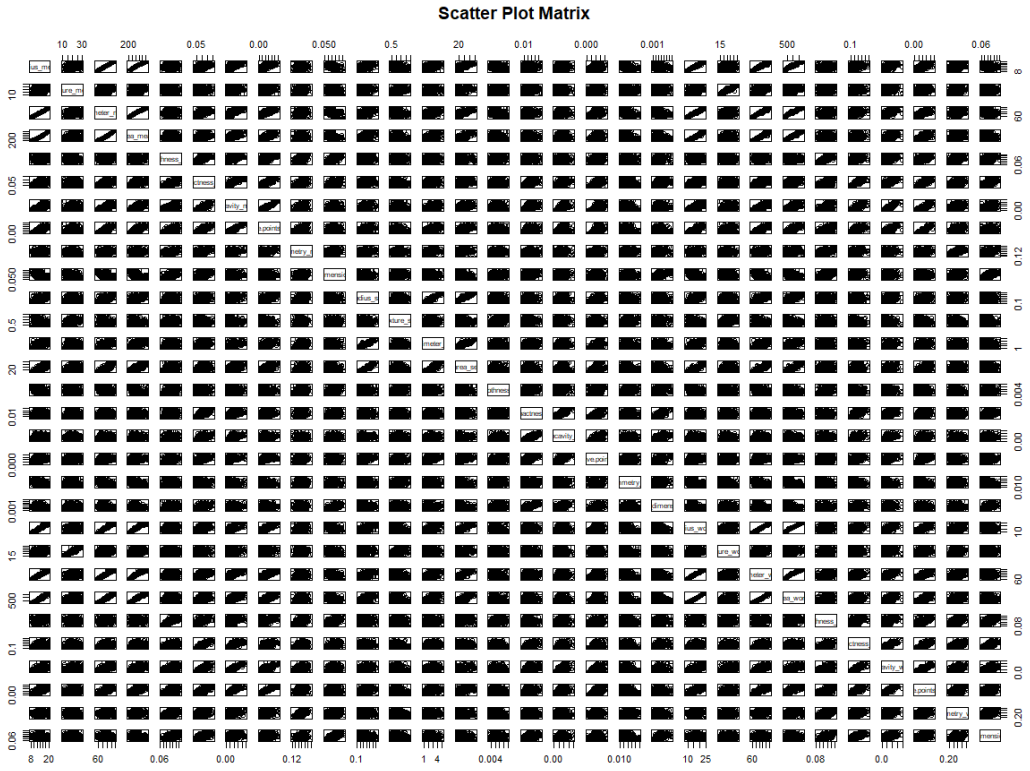
2. 강한 상관관계가 존재하는 변수 조합들 중에 대표 변수를 하나씩만 선택해서 전체변수의 개수를 감소시켜보시오 ([Q7]에서사용함)

[A] 입력변수 조합에 대한 산점도와 correlation plot을 R 을 이용해 그려보았습니다.

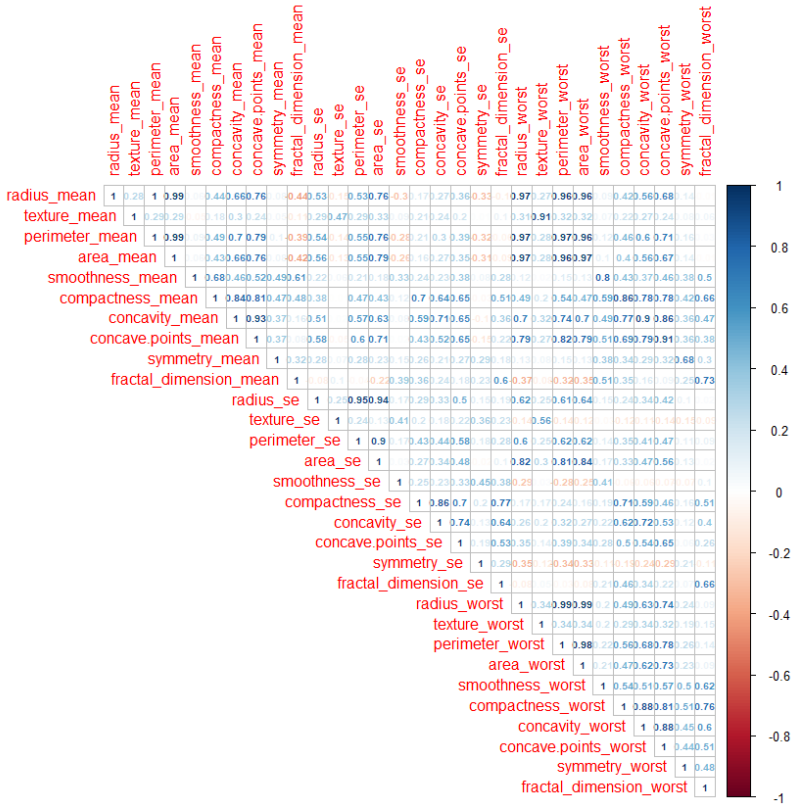
```
#Q5 ScatterPlot 및 Corrplot
par(mar=c(1,1,1,1))
pairs(bc_input_cleared,main="Scatter Plot Matrix")
corrplot(cor(bc_input_cleared), method = 'number', type="upper", number.cex=0.65)
```

-코드-

그 결과 산점도 그래프는 아래의 그림같이 나왔습니다.



변수가 30개나 되어 boxplot을 통해 시각적으로 상관관계를 파악하는 것에는 한계가 있었습니다. 고로 correlation plot을 통해 각 변수들 간의 상관관계가 어떻게 되는지 알아보았습니다.



그 결과 상관관계를 띄는 변수들을 찾아냈습니다. 강한 상관관계의 기준은 관계도의 절대값이 0.9 이상인 것으로 설정하였습니다.
radius mean과 강한 상관관계를 띄는 변수들은 (perimeter_mean, area_mean, radius_worst, perimeter_worst, area_worst)
(6개의 변수는 각각에게만 상호 높은 상관관계를 띔)

texture_mean과 강한 상관관계를 띄는 변수는 texture_ worst
(2개의 변수는 서로에게서만 강한 상관관계를 띔)

concavity_mean과 강한 상관관계를 띄는 변수는 (concave.points_mean, concavity_worst)
(concavity 관련 변수는 셋이서 각각 높은 상관관계를 띔)

radius_se와 강한 상관관계를 띄는 변수는 (perimeter_se, area_se)

perimeter_se와 강한 상관관계를 띄는 변수는 (radius_se)

area_se와 강한 상관관계를 띄는 변수는 (radius_se) 입니다.

이렇게 상관관계를 띄는 그룹을 찾을 수 있다. 이 중 Q7에서 이용하기 위해 제거할 데이터는 위에서 정의한 그룹 안에서 radius_mean과 texture_mean, concavity_mean, radius_se 이렇게 넷을 제외한 나머지 변수들을 제거해주었습니다.

[Q6] 전체 데이터셋을 70%의 학습 데이터와 30%의 테스트 데이터로 분할한 후 모든 변수를 사용하여 Logistic Regression 모델을 학습해 보시오. 이 때 70:30으로 구분하는 random seed를 저장하시오. ([Q7]

1. 유의수준 0.05에서 유효한 변수의 수는 몇 개인지 확인하고 각 변수들이 본인의 상식선에서 실제로 유효하다고 할 수 있는지 판단해 보시오.

2. [Q2-2]에서 정성적으로 선택했던 변수들의 P-value를 확인하고 해당 변수가 모델링 측면에서 실제로 유효하지 않는 것인지 확인해보시오.

3. 학습데이터와 테스트 데이터에 대한 Confusion Matrix를 생성하고 Simple Accuracy, Balanced Correction Rate, F1-Measure를 산출하여 비교해보시오

4. 학습 데이터와 테스트 데이터에 대한 AUROC를 산출하는 함수를 직접 작성하고 이를 사용하여 학습/테스트 데이터셋에 대한 AUROC를 비교해보시오.

```
#Q6 [1~3]
set.seed(12345)
trn_idx <- sample(1:nrow(bc_data_cleared), round(0.7*nrow(bc_data_cleared)))
bc_trn <- bc_data_cleared[trn_idx,]
bc_tst <- bc_data_cleared[-trn_idx,]
# Train the Logistic Regression Model with all variables
full_lr <- glm(bc_target ~ . , family=binomial, bc_trn)
summary(full_lr)
lr_response <- predict(full_lr, type = "response", newdata = bc_tst)
lr_target <- bc_tst$bc_target
lr_predicted <- rep(0, length(lr_target))
lr_predicted[which(lr_response >= 0.5)] <- 1
cm_full <- table(lr_target, lr_predicted)
cm_full
perf_mat[1,] <- perf_eval2(cm_full)
perf_mat
```

-코드-

```
Call:
glm(formula = bc_target ~ ., family = binomial, data = bc_trn)
Deviance Residuals:
      Min         1Q     Median         3Q        Max
-9.599e-05  -2.100e-08  -2.100e-08   2.100e-08   1.046e-04
Coefficients:
                        Estimate Std. Error z value Pr(>|z|)
(Intercept)            -4.988e+01  4.586e+04  -0.001    0.999
radius_mean            -8.224e+02  4.066e+06   0.000    1.000
texture_mean            7.494e+01  1.239e+05   0.001    1.000
perimeter_mean          8.824e+02  2.735e+06   0.000    1.000
area_mean              -2.497e+02  2.802e+06   0.000    1.000
smoothness_mean        -8.089e+00  8.979e+04   0.000    1.000
compactness_mean       -3.117e+02  2.428e+05  -0.001    0.999
concavity_mean          2.501e+02  3.673e+05   0.001    0.999
concave.points_mean     4.876e+01  4.861e+05   0.000    1.000
symmetry_mean           4.971e+00  1.138e+05   0.000    1.000
fractal_dimension_mean  1.457e+02  1.046e+05   0.001    0.999
radius_se               9.719e+01  5.998e+05   0.000    1.000
texture_se             -3.938e+00  5.998e+04   0.000    1.000
perimeter_se            7.451e+01  5.924e+05   0.000    1.000
area_se                -1.969e+02  4.763e+05   0.000    1.000
smoothness_se          -1.016e+02  8.545e+04  -0.001    0.999
compactness_se         -2.858e+01  3.117e+05   0.000    1.000
concavity_se            4.101e+01  4.694e+05   0.000    1.000
concave.points_se       1.664e+02  8.358e+04   0.002    0.998
symmetry_se            -5.354e+01  1.405e+05   0.000    1.000
fractal_dimension_se   -1.088e+02  1.738e+05  -0.001    1.000
radius_worst            1.334e+03  2.758e+06   0.000    1.000
texture_worst           1.541e+00  1.781e+05   0.000    1.000
perimeter_worst        -3.941e+02  1.359e+06   0.000    1.000
area_worst             -5.446e+02  2.353e+06   0.000    1.000
smoothness_worst        7.201e+01  1.266e+05   0.001    1.000
compactness_worst       8.001e+01  2.612e+05   0.000    1.000
concavity_worst         5.350e+00  3.334e+05   0.000    1.000
concave.points_worst   -1.218e+02  3.610e+05   0.000    1.000
symmetry_worst          3.442e+01  1.770e+05   0.000    1.000
fractal_dimension_worst -2.751e+01  1.245e+05   0.000    1.000
(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 3.2678e+02  on 278  degrees of freedom
Residual deviance: 1.1331e-07  on 248  degrees of freedom
AIC: 62
Number of Fisher Scoring iterations: 25
```

```
lr_predicted
lr_target  0  1
       0 89  8
       1 0 22
TPR (Recall) Precision       TNR       ACC       BCR       F1
Logstic Regression 1 0.7333333 0.9175258 0.9327731 0.9578757 0.8461538
```

-코드의 결과-

[A] 주어진 변수들을 모두 이용하여 테스트셋과 트레이닝 셋으로 구분하여 로지스틱 회귀분석 모델을 구한 결과 모든 변수들의 p-val값이 1에 가깝게 나와 모든 변수가 유효하지 않다고 할 수 있다. 당연히 Q2-2에서 정성적으로 선택한 변수들의 p-val 또한 1에 가깝게 나와 유효하지 않다고 할 수 있습니다.

Confusion matrix에서 나온 값을 통해 계산해보면 Simple Accuracy는 0.9327731, Balanced Correction Rate는 0.9578757, F1-Measure는 0.8461538임을 알 수 있습니다.

앞서 말했듯 결과의 p-val로 보아 해당 모델은 전혀 설명력이 없는 모델이여서 30가지의 변수 중 mean 값들로만 모델링을 다시 진행하였습니다.

```r
#reQ6
req6bc_data_cleared<-bc_data_cleared[,c(1,2,3,4,5,6,7,8,9,10,31)]
set.seed(12345)
trn_idx <- sample(1:nrow(req6bc_data_cleared), round(0.7*nrow(req6bc_data_cleared)))
bc_trn <- req6bc_data_cleared[trn_idx,]
bc_tst <- req6bc_data_cleared[-trn_idx,]
# Train the Logistic Regression Model with all variables
full_lr <- glm(bc_target ~ . , family=binomial, bc_trn)
summary(full_lr)
#트레이닝셋 검증
lr_response_train <- predict(full_lr, type = "response", newdata = bc_trn)
lr_target_train  <- bc_trn$bc_target
lr_predicted_train  <- rep(0, length(lr_target_train ))
lr_predicted_train [which(lr_response_train  >= 0.5)] <- 1
cm_full_train  <- table(lr_target_train , lr_predicted_train )
cm_full_train
perf_mat_train <- matrix(0, 1, 6)
colnames(perf_mat_train) <- c("TPR (Recall)", "Precision", "TNR", "ACC", "BCR", "F1")
rownames(perf_mat_train) <- "Logstic Regression"
perf_mat_train[1,] <- perf_eval2(cm_full_train)
perf_mat_train
#테스트셋 검증
lr_response_test <- predict(full_lr, type = "response", newdata = bc_tst)
lr_target_test <- bc_tst$bc_target
lr_predicted_test <- rep(0, length(lr_target_test))
lr_predicted_test[which(lr_response_test >= 0.5)] <- 1
cm_full_test <- table(lr_target_test, lr_predicted_test)
cm_full_test
perf_mat_test <- matrix(0, 1, 6)
colnames(perf_mat_test) <- c("TPR (Recall)", "Precision", "TNR", "ACC", "BCR", "F1")
rownames(perf_mat_test) <- "Logstic Regression"
perf_mat_test[1,] <- perf_eval2(cm_full_test)
perf_mat_test
```

-코드-

```
Call:
glm(formula = bc_target ~ ., family = binomial, data = bc_trn)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.55079  -0.16299  -0.05713   0.00020   2.93644

Coefficients:
                      Estimate Std. Error z value Pr(>|z|)
(Intercept)             1.2076     0.8782   1.375   0.1691
radius_mean            29.1551    26.9779   1.081   0.2798
texture_mean            1.5822     0.3852   4.108    4e-05 ***
perimeter_mean        -36.1241    26.9937  -1.338   0.1808
area_mean               9.3841     8.7212   1.076   0.2819
smoothness_mean         0.2684     0.8458   0.317   0.7510
compactness_mean       -2.5985     2.1401  -1.214   0.2247
concavity_mean          5.2628     2.0319   2.590   0.0096 **
concave.points_mean     4.2703     2.0296   2.104   0.0354 *
symmetry_mean           0.4863     0.6305   0.771   0.4405
fractal_dimension_mean  1.7288     1.0131   1.706   0.0879 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 326.783  on 278  degrees of freedom
Residual deviance:  74.698  on 268  degrees of freedom
AIC: 96.698

Number of Fisher Scoring iterations: 8
```

-Logistic Model 결과-

```
lr_predicted
lr_target   0   1
       0 197   6
       1 8  68
TPR (Recall) Precision       TNR       ACC       BCR        F1
Logstic Regression    0.8947368 0.9189189 0.9704433 0.9498208 0.9318216 0.9066667
```

-트레이닝 셋 결과-

```
lr_predicted
lr_target  0  1
       0 92  5
       1  4 18

TPR (Recall) Precision       TNR       ACC       BCR  F1
Logstic Regression    0.8181818 0.7826087 0.9484536 0.9243697 0.8809129 0.8
```

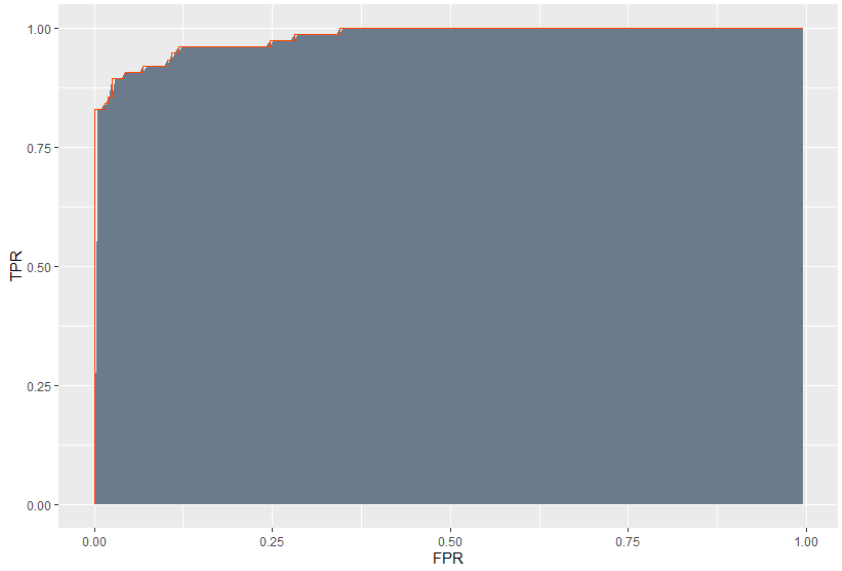
-테스트 셋 결과-

이렇게 다시 변수를 정비한 후 각 변수들의 p-val 을 확인해보았을 때 texture_mean, concavity_mean, concave.points_mean 이 세변수가 유의수준 0.05 에서 유의한 변수인 것을 확인할 수 있습니다. 정성적으로 선택했던 변수들 중 texture_mean, concavity_mean, concave.points_mean 세가지 값이 유의한 변수임을 알 수 있습니다.
Confusion matrix 에서 나온 값을 통해 계산해보면
트레이닝 셋의 결과로는 Simple Accuracy 는 0.9498208,
Balanced Correction Rate 는 0.9318216
F1-Measure 는 0.9066667
테스트 셋의 결과로는 Simple Accuracy 는 0.9243697,
Balanced Correction Rate 는 0.8809129,
F1-Measure 는 0.8 임을 알 수 있습니다.
이 둘의 비교 결과 테스트 셋의 단순정확도와 균형정확도는 트레이닝 셋에 비하여 떨어졌지만 F1 Measure 값이 더 높다는 것을 알 수 있습니다.

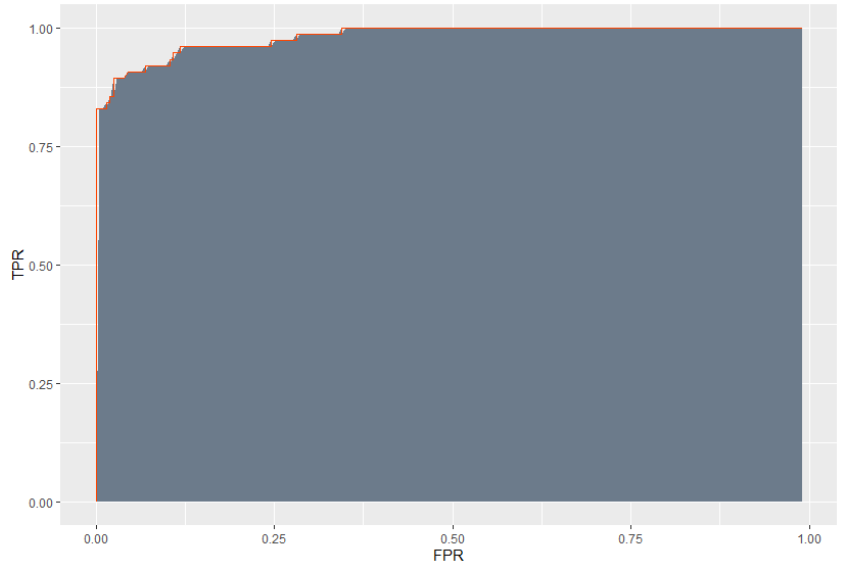
AUROC 는 FPR 과 TPR 값으로 이루어진 그래프로 R 을 이용해 구현해보았습니다.

```
auroc <- function(lr_r,lr_t){
  TPR<-numeric(20000)
  FPR<-numeric(30000)
  aucval=0
  for (i in 1:19999){

    lr_pr<- rep(0, length(lr_t))
    lr_pr[which(lr_r >= i/20000)] <- 1
    cm <- table(lr_t, lr_pr)
    # True positive rate: TPR (Recall)
    TPR[i] <- cm[2,2]/sum(cm[2,])

    # False postitive rate : FPR
    FPR[i] <- cm[1,2]/sum(cm[1,])

    if (i>1) {
      aucval<-aucval+(FPR[i-1]-FPR[i])*(mean(TPR[i],TPR[i-1]))
    }


  }
  TP_FP_D<-data.frame(FPR,TPR)
  ggplot(TP_FP_D,aes(x=FPR,y=TPR))+
    geom_ribbon(aes(ymax=TPR, ymin=0), fill = "slategray4")+
    geom_path(color="orangered")
  print(aucval)
}
auroc(lr_response_train,lr_target_train)
auroc(lr_response_test,lr_target_test)
```

cut-off value 를 미세하게 쪼개어 해당 cut-off value 때마다 TPR 과 FPR 값을 구하여 ROC Curve 를 구현하였습니다. Area Under Curve 즉 AUC 값은 구분구적법을 통하여 값을 구하였다.
Training 데이터의 AUC 값은 0.9714156, Test 데이터의 AUC 값은 0.9695408 이 나왔다.
ROC Curve 는 다음과 같다



-Training 데이터셋의 ROC Curve-



-Test 데이터셋의 ROC Curve-

[Q7] [Q5]에서 변수간 상관관계를 기준으로 선택한 변수들 만을 사용하여 [Q6]에서 사용한 학습/테스트 70:30 분할 데이터로 Logistic Regression 모델을 학습해 보시오.

1. 유의 수준 0.05 에서 유효한 변수의 수는 몇 개인지 확인하고 [Q6-1]의 결과와 비교하시오.
2. 학습 데이터와 테스트 데이터에 대한 Confusion Matrix 를 생성하고 Simple Accuracy, Balanced Correction Rate, F1-Measure 를 산출한 뒤, [Q6-3]의 결과와 비교해 보시오.
3. 학습/테스트 데이터셋에 대한 AUROC 를 산출하여 [Q6-4]의 결과와 비교해 보시오.

[A]

```
#Q7
set.seed(12345)
trn_idx <- sample(1:nrow(q7bc_data_cleared), round(0.7*nrow(q7bc_data_cleared)))
q7bc_trn <- q7bc_data_cleared[trn_idx,]
q7bc_tst <- q7bc_data_cleared[-trn_idx,]
# Train the Logistic Regression Model with all variables
q7full_lr <- glm(bc_target ~ . , family=binomial, q7bc_trn)
summary(q7full_lr)
#학습데이터
q7lr_response_trn <- predict(q7full_lr, type = "response", newdata = q7bc_trn)
q7lr_target <- q7bc_trn$bc_target
q7lr_predicted <- rep(0, length(q7lr_target))
q7lr_predicted[which(q7lr_response_trn >= 0.5)] <- 1
q7cm_full <- table(q7lr_target, q7lr_predicted)
q7cm_full
q7perf_mat_test <- matrix(0, 1, 6)
colnames(q7perf_mat_test) <- c("TPR (Recall)", "Precision", "TNR", "ACC", "BCR", "F1")
rownames(q7perf_mat_test) <- "Logstic Regression"
q7perf_mat_test[1,] <- perf_eval2(q7cm_full)
q7perf_mat_test
#테스트 데이터
q7lr_response_tst <- predict(q7full_lr, type = "response", newdata = q7bc_tst)
q7lr_target <- q7bc_tst$bc_target
q7lr_predicted <- rep(0, length(q7lr_target))
q7lr_predicted[which(q7lr_response_tst >= 0.5)] <- 1
q7cm_full <- table(q7lr_target, q7lr_predicted)
q7cm_full
q7perf_mat_test <- matrix(0, 1, 6)
colnames(q7perf_mat_test) <- c("TPR (Recall)", "Precision", "TNR", "ACC", "BCR", "F1")
rownames(q7perf_mat_test) <- "Logstic Regression"
q7perf_mat_test[1,] <- perf_eval2(cm_full)
q7perf_mat_test
#AUROC
auroc(q7lr_response_trn,lr_target_train)
auroc(q7lr_response_tst,lr_target_test)
```

-코드-

```
Call:
glm(formula = bc_target ~ ., family = binomial, data = q7bc_trn)
Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.81957  -0.19235  -0.06032   0.00107   2.95089
Coefficients:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)           0.1852     0.4036   0.459  0.64622
radius_mean           4.4818     1.0861   4.127 3.68e-05 ***
texture_mean          1.4964     0.3612   4.143 3.43e-05 ***
smoothness_mean       1.5368     0.6804   2.259  0.02390 *
compactness_mean     -3.7379     1.3992  -2.671  0.00755 **
concavity_mean        5.8950     1.4324   4.115 3.87e-05 ***
symmetry_mean         0.1549     0.5715   0.271  0.78630
fractal_dimension_mean  1.1986     0.9036   1.326  0.18471
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 326.783  on 278  degrees of freedom
Residual deviance:  82.091  on 271  degrees of freedom
AIC: 98.091
Number of Fisher Scoring iterations: 8
```

-결과-

Q5 에서 골라낸 변수와 Q6-2 에서 다시 골라낸 변수들을 종합해, radius_mean, texture_mean, smoothness_mean, compactness_mean, concavity_mean, symmetry_mean, fracta_dimension_mean 이렇게 7 가지 설명변수만을 선정하여 로지스틱 회귀분석 모델을 구하였습니다.
그 결과 유의수준 0.05 기준, radius_mean, texture_mean, smoothness_mean, compactness_mean, concavity_mean 총 5 개의 변수가 유의한 변수인 것을 확인하였습니다.

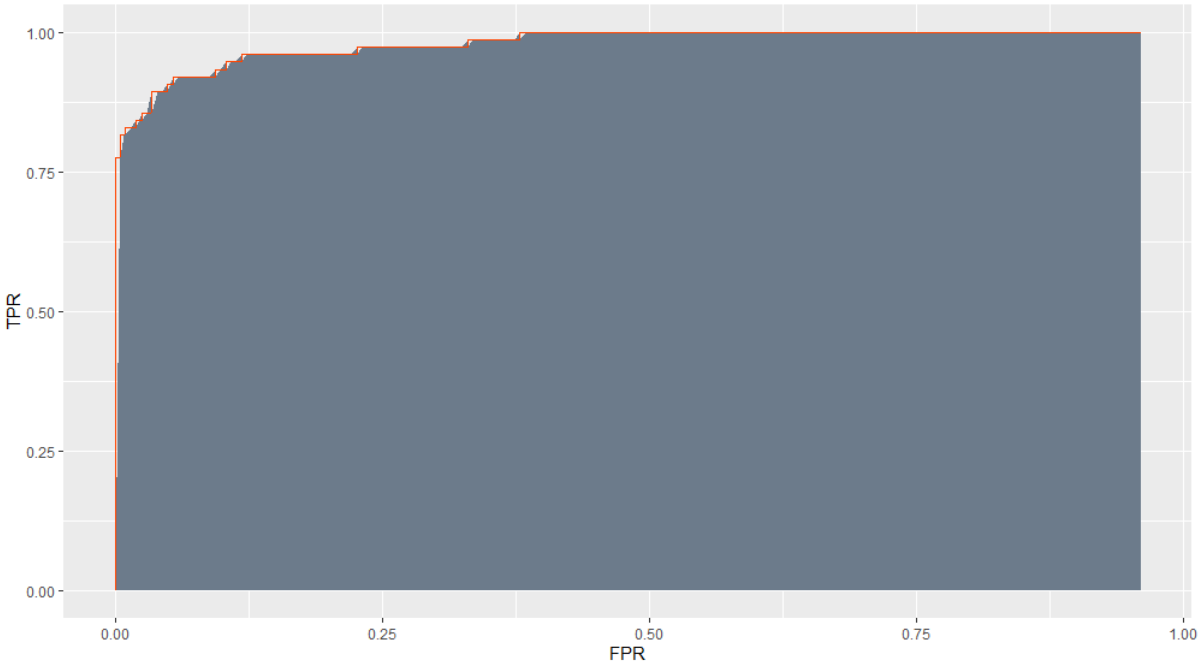
```
TPR (Recall) Precision       TNR       ACC       BCR        F1
Logstic Regression    0.8552632 0.9027778 0.9655172 0.9354839 0.9087196 0.8783784
```

-트레이닝 셋-

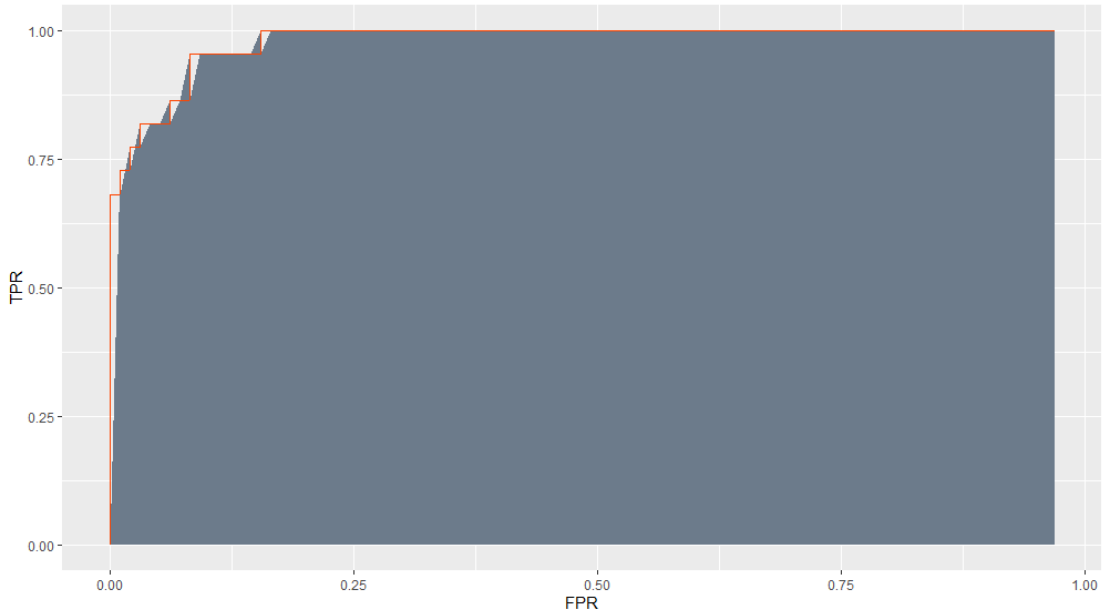
```
TPR (Recall) Precision       TNR       ACC       BCR        F1
Logstic Regression    0.8636364  0.826087 0.9587629 0.9411765 0.9099574 0.8444444
```

-테스트 셋-

Confusion matrix 에서 나온 값을 통해 계산해보면
트레이닝 셋의 결과로는 Simple Accuracy 는 0.9354839,
Balanced Correction Rate 는 0.9087196
F1-Measure 는 0.8783784
테스트 셋의 결과로는 Simple Accuracy 는 0.9411765,
Balanced Correction Rate 는 0.9099574,
F1-Measure 는 0.8444444 임을 알 수 있습니다.
이 둘의 비교 결과 테스트 셋의 단순정확도와 균형정확도,F1 Measure 값 모두 트레이닝 셋보다 높다는 것을 알 수 있습니다.



-트레이닝 셋의 ROC Curve-



-테스트 셋의 ROC Curve-

Training 데이터의 AUC 값은 0.9404978, Test 데이터의 AUC 값은 0.9732896 이 나왔다. ROC Curve 는 다음과 같다.