Predicting Formation of Calcium Oxalate Crystals in Urine

Parikshit Patil

2024-09-01

Predicting Formation of Calcium Oxalate Crystals in Urine

```
suppressMessages(library(boot))
which(is.na(urine),arr.ind = TRUE)
```

```
##    row col
## 55  55   4
## 1    1   5
```

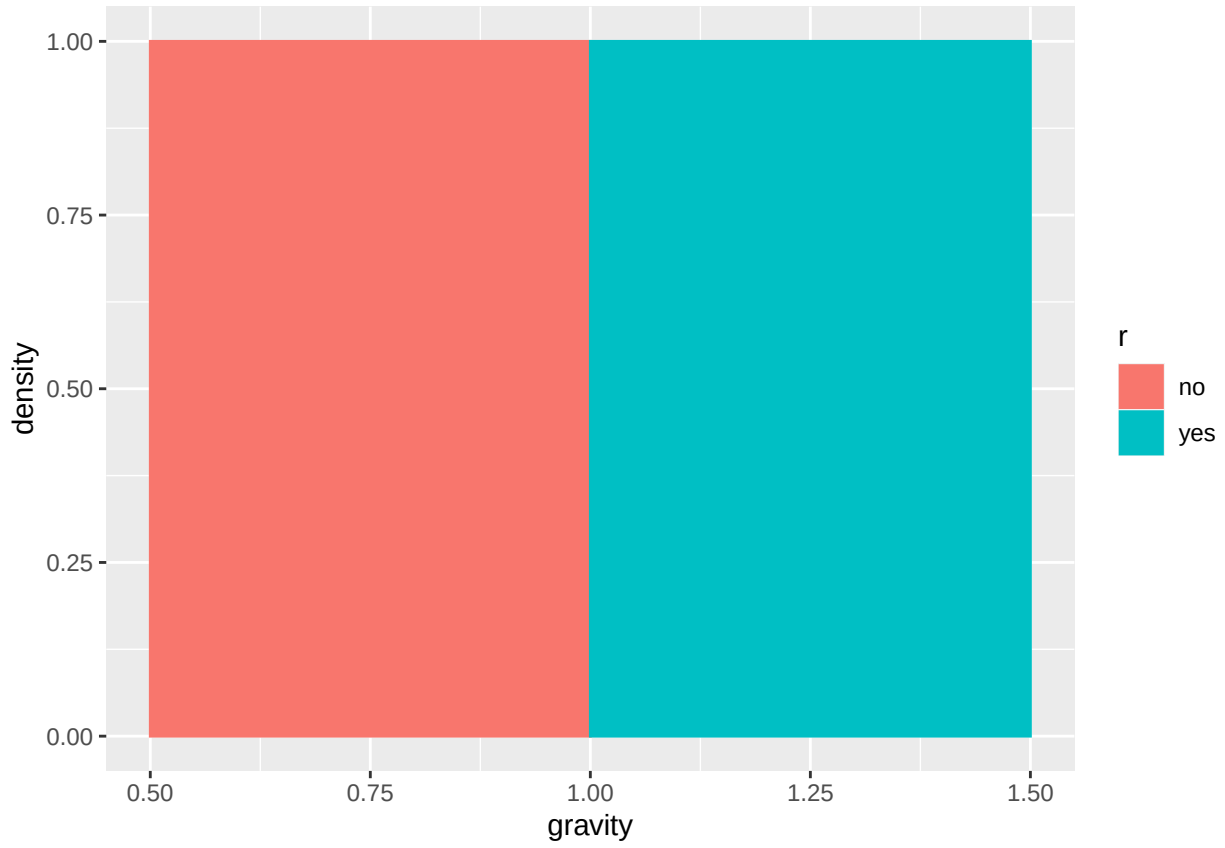
```
urine <- urine[-c(1,55),]
urine$r <- factor(urine$r, levels= c("0","1"),labels = c("no","yes"))
str(urine)
```

```
## 'data.frame':    77 obs. of  7 variables:
##  $ r      : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ gravity: num  1.02 1.01 1.01 1 1.02 ...
##  $ ph     : num  5.74 7.2 5.51 6.52 5.27 5.62 5.67 5.41 6.13 6.19 ...
##  $ osmo   : num  577 321 408 187 668 ...
##  $ cond   : num  20 14.9 12.6 7.5 25.3 17.4 35.9 21.9 25.7 11.5 ...
##  $ urea   : num  296 101 224 91 252 195 550 170 382 152 ...
##  $ calc   : num  4.49 2.36 2.15 1.16 3.34 1.4 8.48 1.16 2.21 1.93 ...
```

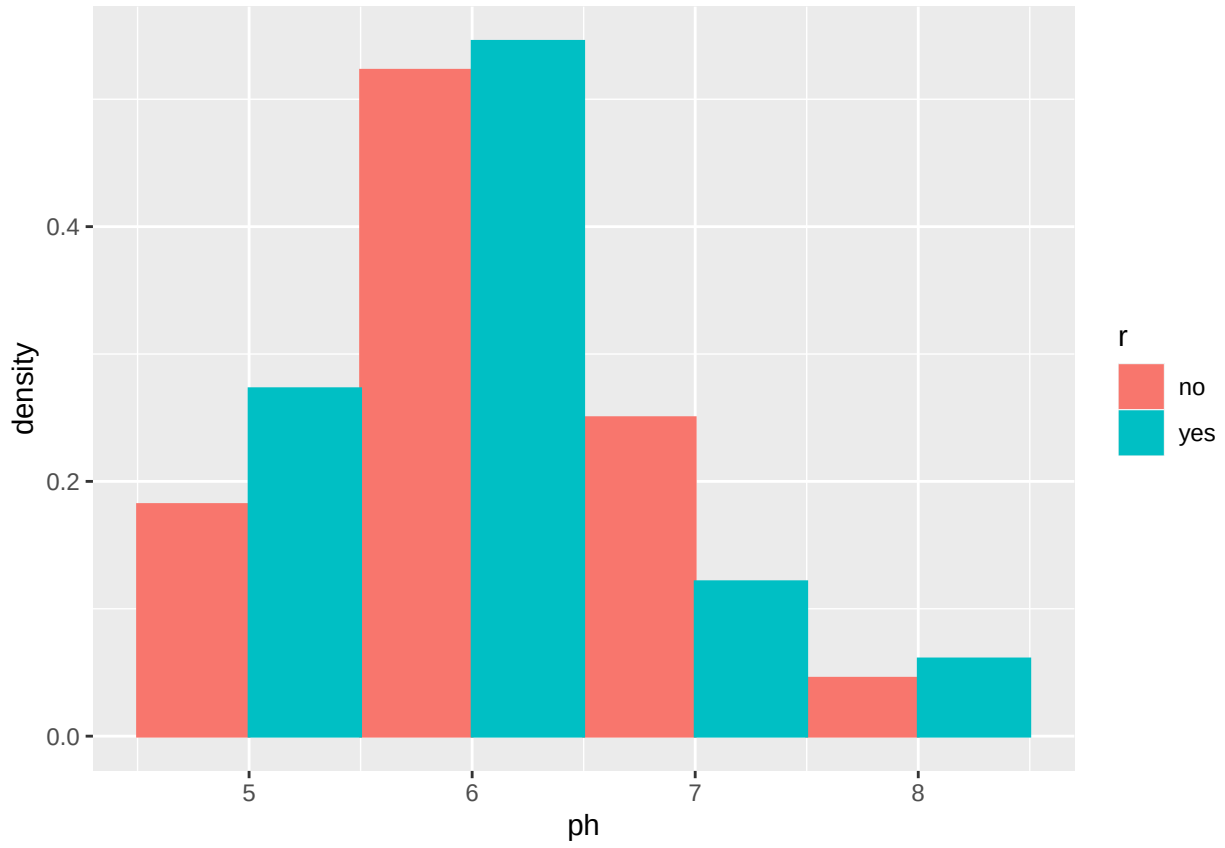
a)

```
#?urine

library(ggplot2)
```

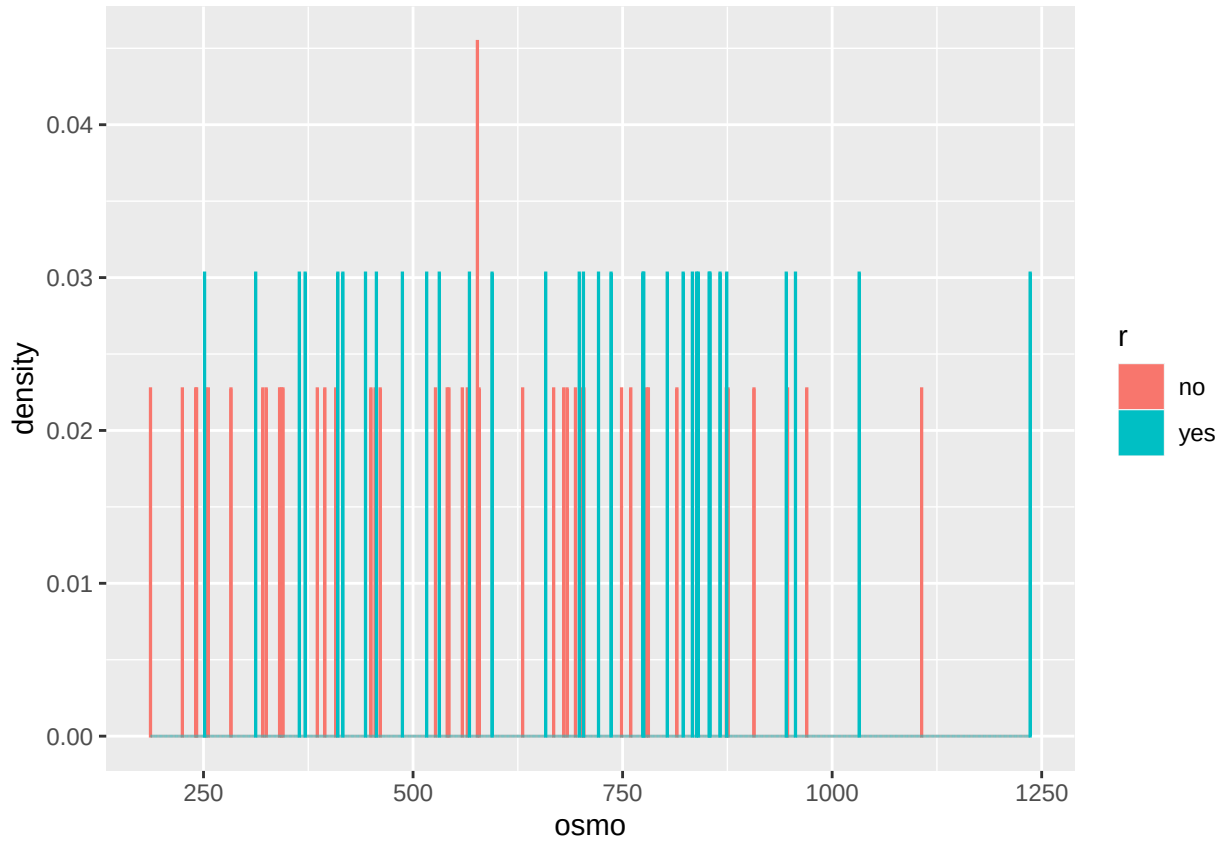
```
#histogram plots for each predictor(covariate)

ggplot(urine, aes(x=gravity, fill=r, color=r)) +
  geom_histogram(position="dodge", binwidth=1, aes(y=after_stat(density)))
```



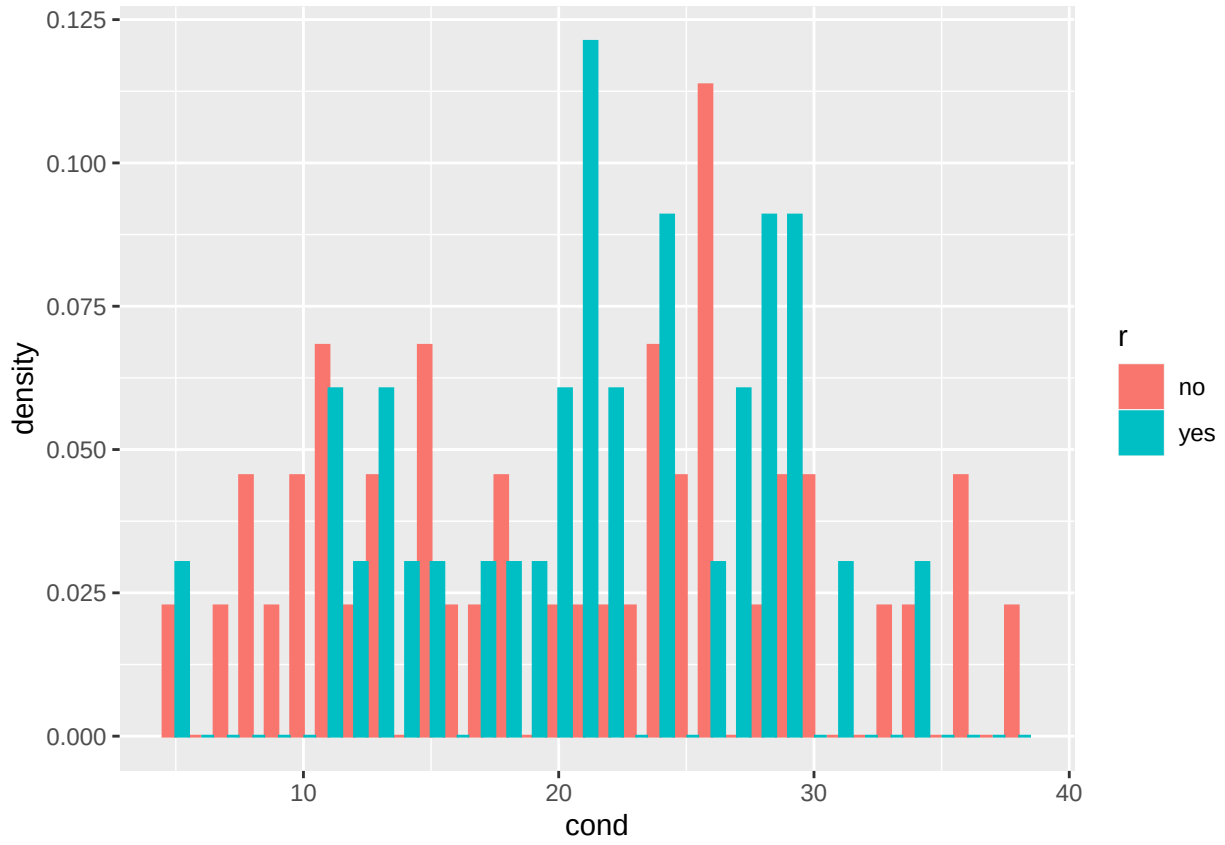

```
  #theme(text = element_text(size = 22))

ggplot(urine, aes(x=ph, fill=r, color=r)) +
  geom_histogram(position="dodge", binwidth=1, aes(y=after_stat(density)))
```



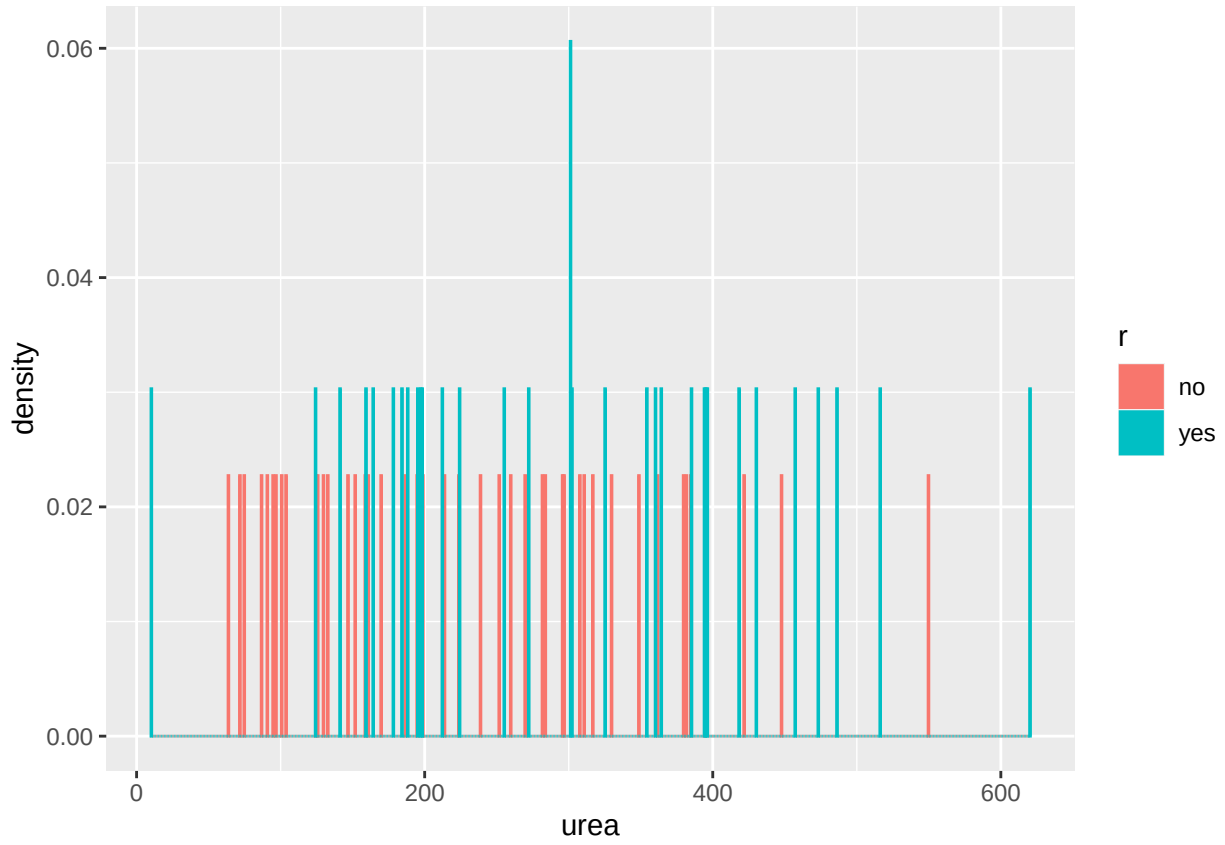

```
ggplot(urine, aes(x=osmo, fill=r, color=r)) +
  geom_histogram(position="dodge", binwidth=1, aes(y=after_stat(density)))
```



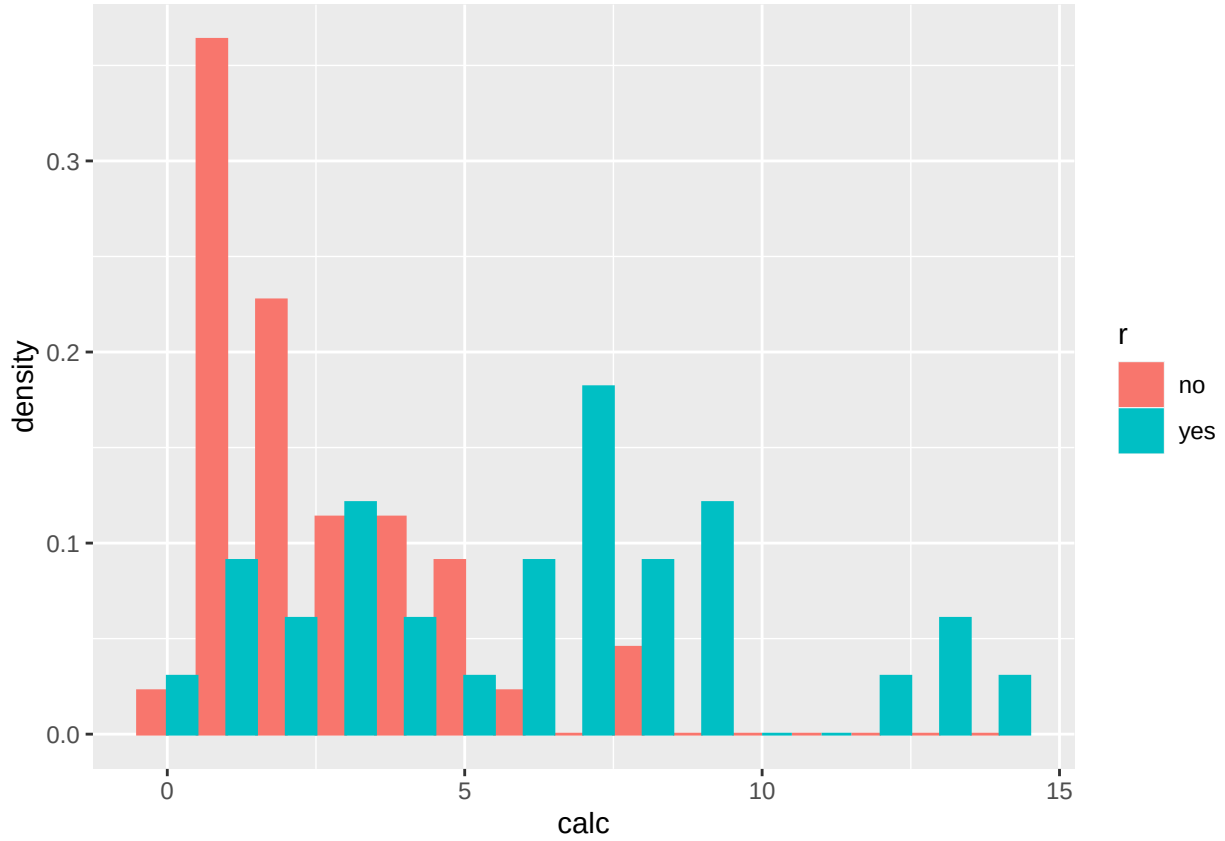

```
ggplot(urine, aes(x=cond, fill=r, color=r)) +
  geom_histogram(position="dodge", binwidth=1, aes(y=after_stat(density)))
```




```
ggplot(urine, aes(x=urea, fill=r, color=r)) +
  geom_histogram(position="dodge", binwidth=1, aes(y=after_stat(density)))
```




```
ggplot(urine, aes(x=calc, fill=r, color=r)) +
  geom_histogram(position="dodge", binwidth=1, aes(y=after_stat(density)))
```



The covariate "calc": seems most likely to be useful in predicting formation of calcium oxalate crystals "r", because through the graphs it can be clearly observed that the ratio of having r vs not increases as calc increases.

b)

```
#Fit full logistic model
logistic_model = glm(r ~ ., family = binomial, data = urine)
summary(logistic_model)
```

```
##
## Call:
## glm(formula = r ~ ., family = binomial, data = urine)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -355.33771  222.76696  -1.595  0.11069
## gravity      355.94379  222.11004   1.603  0.10903
## ph            -0.49570    0.56976  -0.870  0.38429
## osmo           0.01681    0.01782   0.944  0.34536
## cond          -0.43282    0.25123  -1.723  0.08493 .
## urea          -0.03201    0.01612  -1.986  0.04703 *
## calc           0.78369    0.24216   3.236  0.00121 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 105.17  on 76  degrees of freedom
## Residual deviance:  57.56  on 70  degrees of freedom
## AIC: 71.56
##
## Number of Fisher Scoring iterations: 6
```

Residual Deviance = 57.56, Degrees of Freedom = 70

This information alone is insufficient to assess if the model fits the data well, as we can show (via L'Hôpital's rule) that each term in log LSat is zero and the deviance depends only on log LM.

c)

```
#p-value calculation for hypothesis test

difference = logistic_model$null.deviance - logistic_model$deviance
#difference

#p_value = 1-pchisq(difference,2)
p_value = pchisq(difference,6,lower.tail = FALSE)
p_value
```

```
## [1] 1.415118e-08
```

The Null Hypothesis is that none of the predictors (null model) are related to the response variable 'r'.

The Alternate Hypothesis is that at least one of the predictors (from full model) are related to the response variable 'r'.

The Test Statistic is the difference of the deviance between the null model (smaller model s) and the full model (larger model l). Test Statistic = 47.6079

$$Test\ Statisitic\ D_s - D_L = 47.6079$$

The asymptotic distribution is the values of the chi-squared difference between the full model (larger model l) and null model (smaller model s).

$$Asymptotic\ Distribution = \chi^2_{l-s}$$

Since the p-value is very small (<0.5), the conclusion is that there exists at least one predictor that is related to the response variable 'r'. We reject the null hypothesis and assume the alternate hypothesis to be true.

$$P\ Value = 1.415118e - 08$$

d)

```
#AIC Feature Selection

logistic_model_reduced = step(logistic_model, trace=0)
summary(logistic_model_reduced)
```

```
##
## Call:
## glm(formula = r ~ gravity + cond + urea + calc, family = binomial,
##     data = urine)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -500.01090  161.87095  -3.089  0.00201 **
## gravity      497.12038  161.32939   3.081  0.00206 **
## cond          -0.20547    0.07105  -2.892  0.00383 **
## urea          -0.01783    0.00723  -2.466  0.01367 *
## calc           0.72232    0.21997   3.284  0.00102 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 105.168  on 76  degrees of freedom
## Residual deviance:  59.071  on 72  degrees of freedom
## AIC: 69.071
##
## Number of Fisher Scoring iterations: 6
```

Using the backward AIC process, the best subset of predictors are: 'gravity', 'cond', 'urea', and 'calc'.

e)

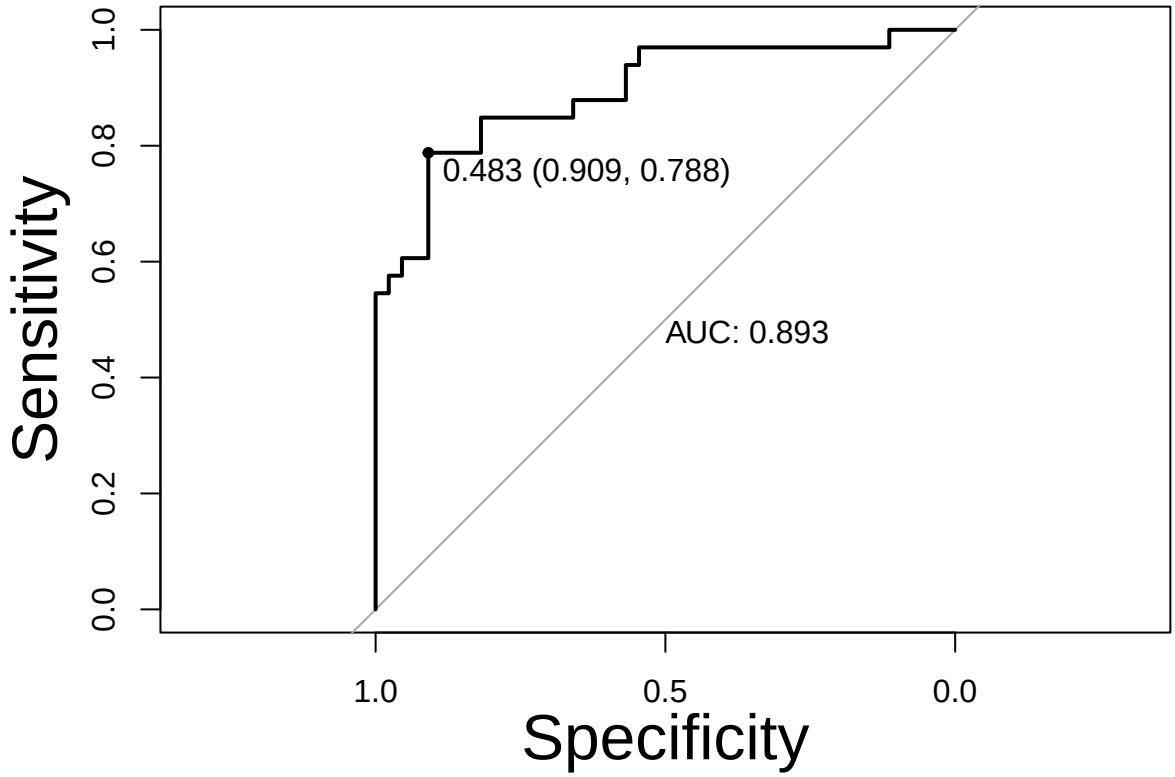
```
library(pROC)
```

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

##
## Attaching package: 'pROC'

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

```
predicted_probabilities = predict(logistic_model_reduced, type = "response")
roc_obj = roc(response = urine$r, predictor = predicted_probabilities)
```

```
## Setting levels: control = no, case = yes

## Setting direction: controls < cases
```

```
#ROC Curve
plot(roc_obj,legacy.axes=FALSE,print.auc=TRUE,print.thres=TRUE,cex.lab=2)
```




```
#AUC
AUC = auc(roc_obj)
AUC
```

```
## Area under the curve: 0.8933
```

```
roc_logistic = c(coords(roc_obj, "b",
  ret=c("threshold","se","sp"),
  best.method="youden"))

#Threshold, Sensitivity, Specificity values
names(roc_logistic) <- c("Threshold","Sensitivity","Specificity")

t(roc_logistic)
```

```
##      Threshold Sensitivity Specificity
## [1,] 0.4830697 0.7878788   0.9090909
```

Area under the curve = 0.8933

The best probability threshold = 0.4830697

Its corresponding sensitivity = 0.7878788, and specificity = 0.9090909

f)

```r
#Confusion Matrix based on best probability threshold found previously (0.48)

cutoff = 0.48
y_hat = numeric(nrow(urine))
y_hat[which(predicted_probabilities > cutoff)] = 1
conf_mat = table(predicted = y_hat, actual = urine$r)

conf_mat
```

```
##          actual
## predicted no yes
##         0 40   7
##         1  4  26
```

```r
FPR = 1 - conf_mat[1, 1] / sum(conf_mat[, 1]) # False Positive Rate
TPR = conf_mat[2, 2] / sum(conf_mat[, 2]) # True Positive Rate
PPV = conf_mat[2, 2] / sum(conf_mat[2, ]) # Positive Predictive Value
NPV = conf_mat[1, 1] / sum(conf_mat[1, ]) # Negative Predictive Value

FPR
```

```
## [1] 0.09090909
```

```r
TPR
```

```
## [1] 0.7878788
```

```r
PPV
```

```
## [1] 0.8666667
```

```r
NPV
```

```
## [1] 0.8510638
```

False Positive Rate (FPR) = 0.09090909

True Positive Rate (TPR) = 0.7878788

Positive Predictive Value (PPV) = 0.8666667

Negative Predictive Value (NPV) = 0.8510638

g)

The model seems effective in predicting the presence of oxilate crystals in the urine, since the TPR, PPV, and NPV are high, especially the AUC. However, we also want to have a higher FPR for healthcare use cases.

h)

```r
# Train-Test Split
set.seed(10)
train_ind <- sample(1:77,51)
train_urine <- urine[train_ind,]
test_urine <- urine[-train_ind,]
```

```r
#Fit the full model
logistic_model = glm(r ~ ., family = binomial, data = train_urine)

#Stepwise Feature Selection on full model
logistic_model_best = step(logistic_model, trace=0)
summary(logistic_model_reduced)
```

```
##
## Call:
## glm(formula = r ~ gravity + cond + urea + calc, family = binomial,
##     data = urine)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -500.01090  161.87095  -3.089  0.00201 **
## gravity      497.12038  161.32939   3.081  0.00206 **
## cond          -0.20547    0.07105  -2.892  0.00383 **
## urea          -0.01783    0.00723  -2.466  0.01367 *
## calc           0.72232    0.21997   3.284  0.00102 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 105.168  on 76  degrees of freedom
## Residual deviance:  59.071  on 72  degrees of freedom
## AIC: 69.071
##
## Number of Fisher Scoring iterations: 6
```

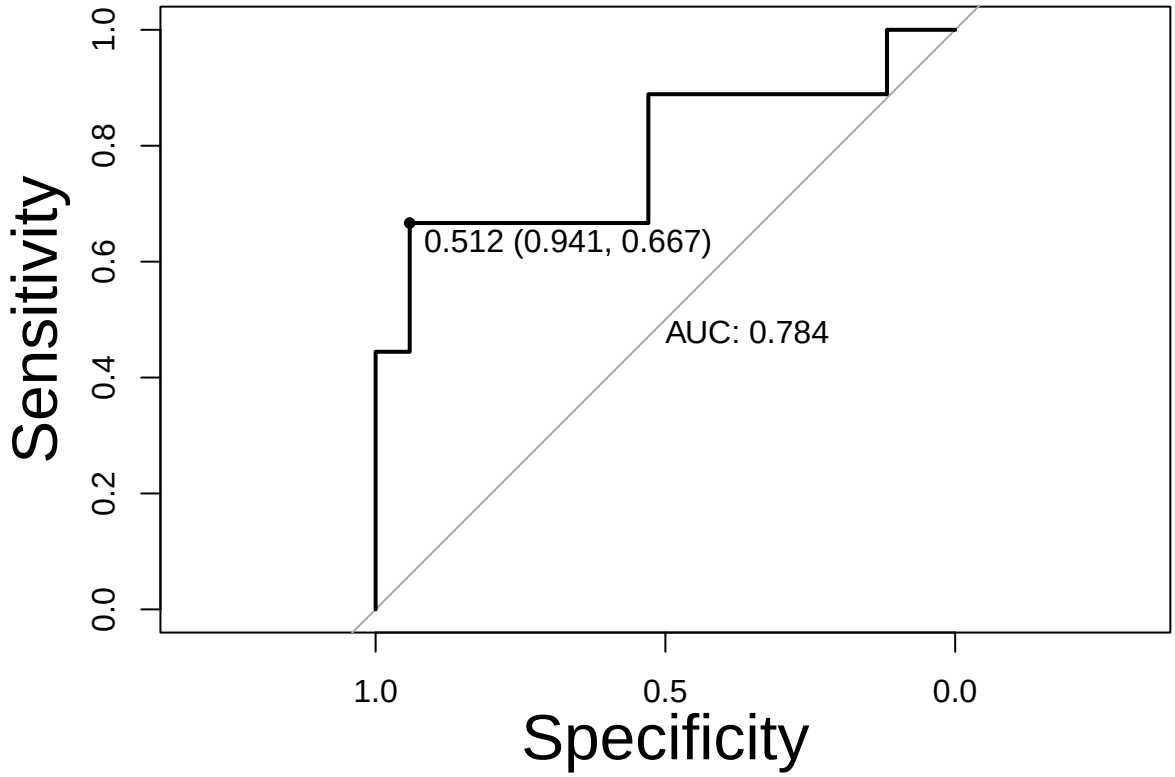
Using only the training data to determine the best model, the remaining variables are: 'gravity', 'cond', 'urea', and 'calc'. This is the same as in d).

```r
predicted_probabilities = predict(logistic_model_best, newdata = test_urine, type = "response")
roc_obj = roc(response = test_urine$r, predictor = predicted_probabilities)
```

```
## Setting levels: control = no, case = yes
```

```
## Setting direction: controls < cases
```

```r
#ROC Curve
plot(roc_obj,legacy.axes=FALSE,print.auc=TRUE,print.thres=TRUE,cex.lab=2)
```




```
#AUC
AUC = auc(roc_obj)
AUC
```

```
## Area under the curve: 0.7843
```

```
#Threshold, Sensitivity, Specificity
roc_logistic = c(coords(roc_obj, "b",
  ret=c("threshold","se","sp"),
  best.method="youden"))

names(roc_logistic) <- c("Threshold","Sensitivity","Specificity")

t(roc_logistic)
```

```
##      Threshold Sensitivity Specificity
## [1,] 0.5115104 0.6666667   0.9411765
```

Area under the curve = 0.7843

The best probability threshold = 0.5115104

Its corresponding sensitivity = 0.6666667, and specificity = 0.9411765

This result is not as good as the previous model because the AUC of this model is lower.