BST260 - Final project

Sabine Friedrich

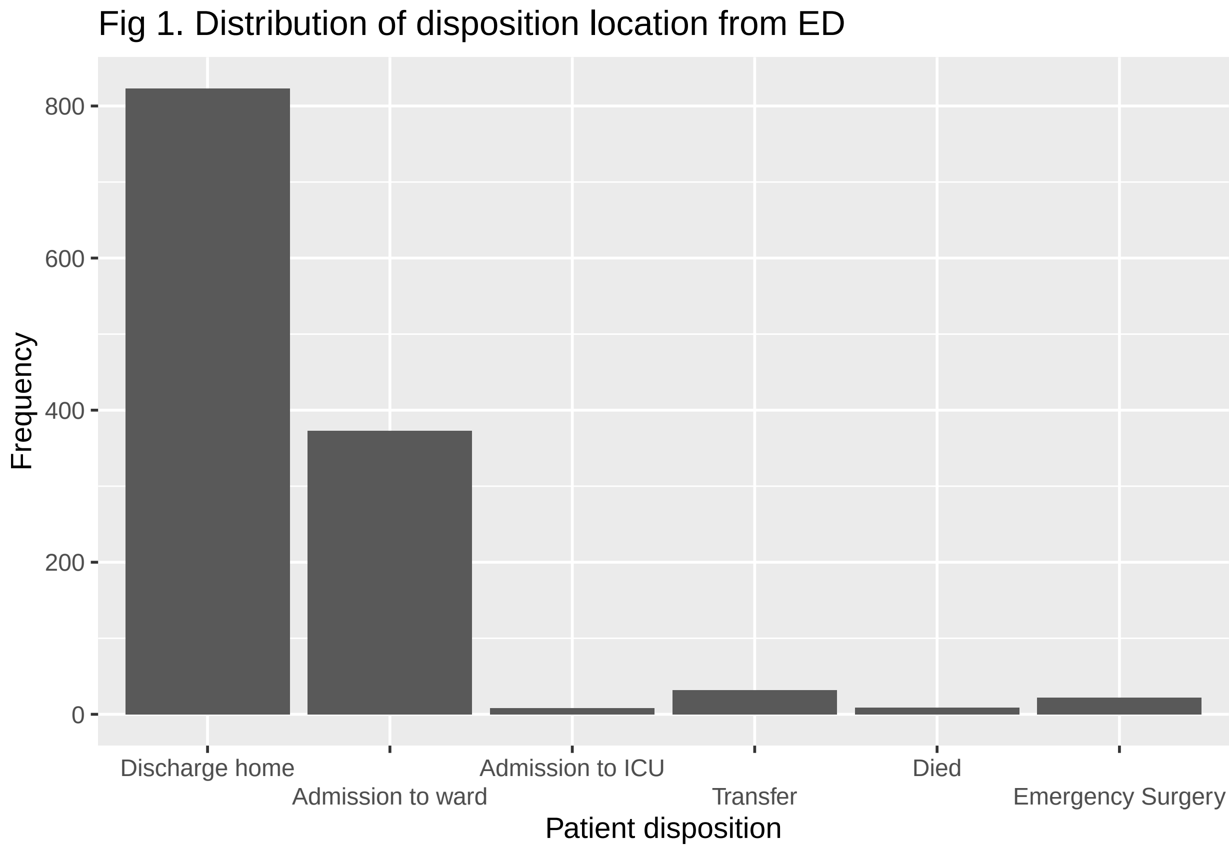
12/15/2022

Introduction

The dataset that I will analyze was assembled by Korean investigators for a cross-sectional retrospective research study aiming to evaluate accuracy of triage in the emergency department by the Korean Triage and Acuity Scale. The original study report was published in 2019 (1) and the dataset was made available on kaggle.com. This is a tidy dataset including 1267 records of adult patients who were admitted to the emergency department (ED) at two different hospitals between October 2016 and September 2017. It includes a variable detailing the disposition of each patient upon discharge from the ED. My initial plan to predict emergency surgery aiming to identify patients who may require emergency surgery early in order to reduce the time until start of the surgical procedure. However, there were only 22 patients (1.7%) who required emergency surgery upon exploratory analysis (Fig. 1). Accordingly, the aim of this project was adapted to predict inpatient admission (including mortality, or transfer to another hospital) in contrast to discharge home. The ability to predict hospital admission of ED patients may help guide and refine the triage process.

```
##   1   2   3   4   5   6   7
## 797 373   8  26  32   9  22
```

```
##   1   2   3   5   6   7
## 823 373   8  32   9  22
```




```
##   0   1
## 823 444
```

To identify predictors of inpatient admission, I will compare two approaches:

- Clinical approach: pre-select candidate predictors based on clinical reasoning and expertise and then use an automated selection process to build and refine a regression model
- Machine learning approach:

Model discrimination will be evaluated using overall accuracy, sensitivity, specificity and AUC. Model calibration will be assessed using a calibration plot comparing predicted probability and observed rates across deciles of predicted risk.

Results

- 6+ key plots or tables illustrating your two major analyses guide the reader through your analysis and describe what each plot or table is showing, and how it relates to the central question you are trying to ask

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##    50.0   114.0   130.0   133.6   150.0   275.0      25 
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   31.00   70.00   80.00   79.78   90.00  160.00      29 
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   32.00   72.00   82.00   83.96   96.00  148.00      20 

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   14.00   18.00   20.00   19.51   20.00   30.00      22 

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   35.00   36.20   36.50   36.58   36.80   41.00      18 

##   1   2   3   4 
##  79 421 753  14 

##    0    1 
## 1023  244 

##    0    1    2    3    4    5    6    7    8    9   10 NA's 
##  553    2   38  278  141  136   70   33    9    1    3    3 

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##  0.2286  0.5221  0.6375  0.6525  0.7455  1.5934      25 

##    0    1 NA's 
## 1197   45   25 

##    0    1 NA's 
## 1197   45   25 
```

All cases with any missing values for potential predictors were removed and a total of 1228 cases remained in the complete case cohort. This cohort was split into a training dataset (80% of observations) and a validation set (remaining 20%).

```
## 
##    0    1  Sum 
##  810  418 1228 

## 
##         0         1 
## 0.6596091 0.3403909 

## 
##   0   1 Sum 
## 669 313 982 

## 
##         0         1 
## 0.6812627 0.3187373 
```

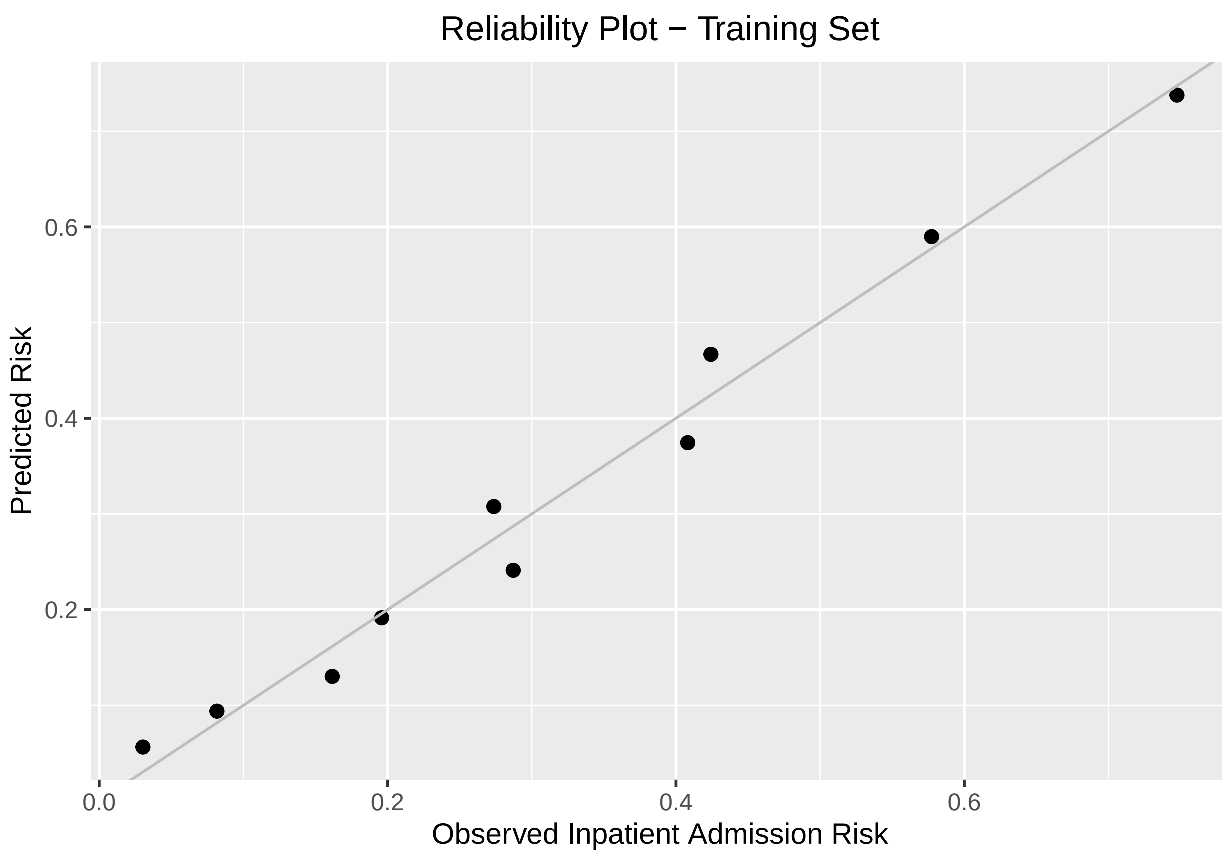
```
## Start:  AIC=1231.3
## admission ~ 1
##
##                       Df Deviance    AIC
## + KTAS_RN              1   1093.9 1097.9
## + as.factor(arrival)  3   1147.3 1155.3
## + Age                  1   1177.8 1181.8
## + injury               1   1197.2 1201.2
## + hypertherm           1   1218.0 1222.0
## + shock                1   1220.7 1224.7
## + NRS_pain             1   1221.7 1225.7
## + hyperventilation     1   1224.4 1228.4
## + mental_status        1   1226.4 1230.4
## + as.factor(Sex)       1   1226.7 1230.7
## <none>                     1229.3 1231.3
##
## Step:  AIC=1097.88
## admission ~ KTAS_RN
##
##                       Df Deviance    AIC
## + Age                  1   1064.2 1070.2
## + as.factor(arrival)  3   1062.3 1072.3
## + injury               1   1081.4 1087.4
## + hypertherm           1   1083.4 1089.4
## + shock                1   1090.1 1096.1
## + NRS_pain             1   1090.6 1096.6
## <none>                     1093.9 1097.9
## + as.factor(Sex)       1   1092.7 1098.7
## + hyperventilation     1   1093.2 1099.2
## + mental_status        1   1093.7 1099.7
##
## Step:  AIC=1070.2
## admission ~ KTAS_RN + Age
##
##                       Df Deviance    AIC
## + as.factor(arrival)  3   1040.0 1052.0
## + hypertherm           1   1051.9 1059.9
## + injury               1   1056.3 1064.3
## + shock                1   1059.5 1067.5
## + as.factor(Sex)       1   1062.2 1070.2
## <none>                     1064.2 1070.2
## + NRS_pain             1   1063.3 1071.3
## + mental_status        1   1063.7 1071.7
## + hyperventilation     1   1063.7 1071.7
##
## Step:  AIC=1052.02
## admission ~ KTAS_RN + Age + as.factor(arrival)
##
##                     Df Deviance    AIC
## + hypertherm         1   1027.9 1041.9
## + injury             1   1029.7 1043.7
## + shock              1   1034.6 1048.6
## + mental_status      1   1037.6 1051.6
## <none>                   1040.0 1052.0
```

```
## + as.factor(Sex)    1   1039.2 1053.2
## + NRS_pain          1   1039.5 1053.5
## + hyperventilation  1   1039.7 1053.7
##
## Step:  AIC=1041.89
## admission ~ KTAS_RN + Age + as.factor(arrival) + hypertherm
##
##                    Df Deviance    AIC
## + injury            1   1019.5 1035.5
## + shock             1   1023.0 1039.0
## + mental_status     1   1024.4 1040.4
## <none>                  1027.9 1041.9
## + as.factor(Sex)    1   1027.2 1043.2
## + NRS_pain          1   1027.6 1043.6
## + hyperventilation  1   1027.9 1043.9
##
## Step:  AIC=1035.55
## admission ~ KTAS_RN + Age + as.factor(arrival) + hypertherm +
##     injury
##
##                    Df Deviance    AIC
## + shock             1   1014.8 1032.8
## + mental_status     1   1016.7 1034.7
## <none>                  1019.5 1035.5
## + as.factor(Sex)    1   1018.4 1036.4
## + NRS_pain          1   1019.5 1037.5
## + hyperventilation  1   1019.5 1037.5
##
## Step:  AIC=1032.83
## admission ~ KTAS_RN + Age + as.factor(arrival) + hypertherm +
##     injury + shock
##
##                    Df Deviance    AIC
## + mental_status     1   1011.6 1031.7
## <none>                  1014.8 1032.8
## + as.factor(Sex)    1   1013.6 1033.6
## + hyperventilation  1   1014.8 1034.8
## + NRS_pain          1   1014.8 1034.8
##
## Step:  AIC=1031.65
## admission ~ KTAS_RN + Age + as.factor(arrival) + hypertherm +
##     injury + shock + mental_status
##
##                    Df Deviance    AIC
## <none>                  1011.6 1031.7
## + as.factor(Sex)    1   1010.2 1032.2
## + NRS_pain          1   1011.6 1033.6
## + hyperventilation  1   1011.6 1033.7
```

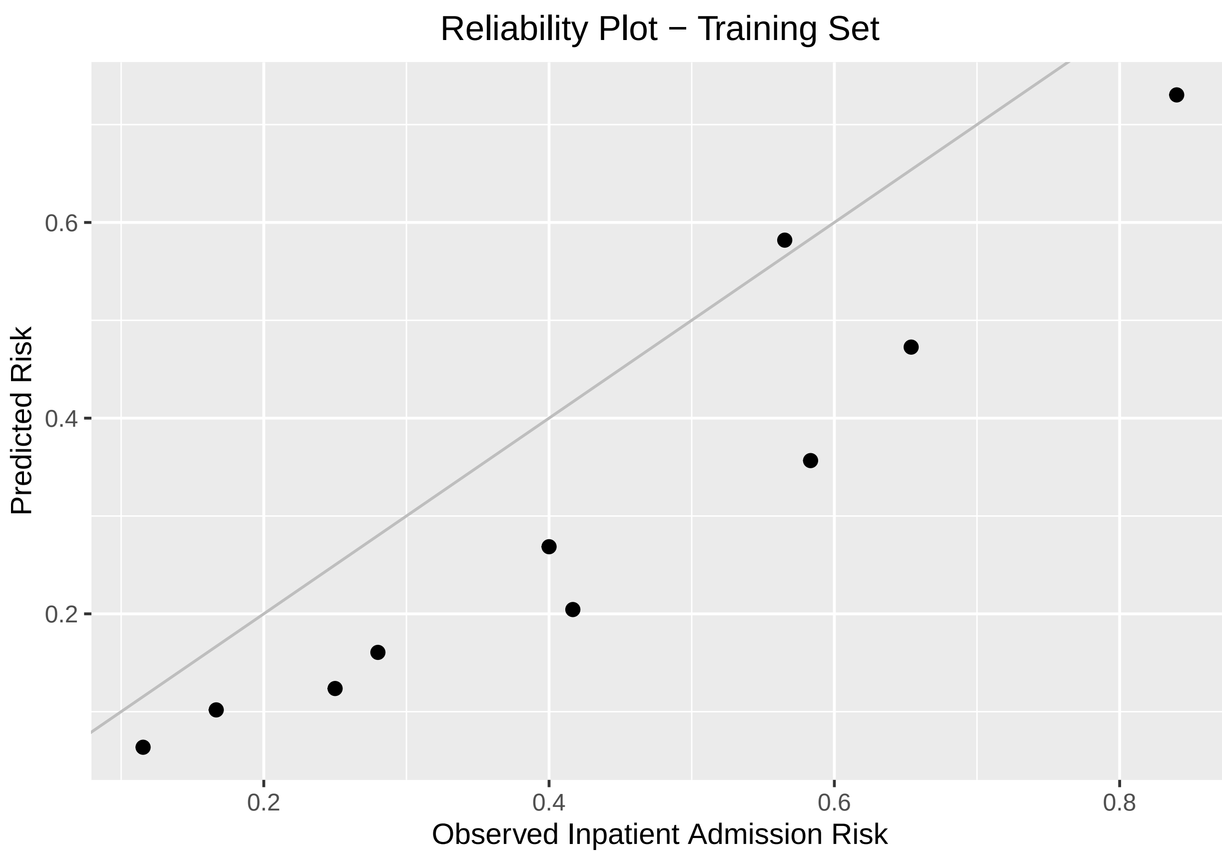
```
##
## Call:
## glm(formula = admission ~ KTAS_RN + Age + as.factor(arrival) +
##     hypertherm + injury + shock + mental_status, family = "binomial",
##     data = ED.train)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8111  -0.7988  -0.4736   0.9040   2.4661
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)          0.374504   0.622920   0.601 0.547702
## KTAS_RN             -0.790070   0.103579  -7.628 2.39e-14 ***
## Age                  0.019035   0.004234   4.496 6.93e-06 ***
## as.factor(arrival)2  1.362495   0.400187   3.405 0.000663 ***
## as.factor(arrival)3  0.572665   0.391449   1.463 0.143484
## as.factor(arrival)4 -0.258425   0.879563  -0.294 0.768903
## hypertherm           1.100834   0.337552   3.261 0.001109 **
## injury              -0.643295   0.242522  -2.653 0.007989 **
## shock                0.875347   0.388884   2.251 0.024391 *
## mental_status       -0.426530   0.239006  -1.785 0.074326 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1229.3  on 981  degrees of freedom
## Residual deviance: 1011.6  on 972  degrees of freedom
## AIC: 1031.6
##
## Number of Fisher Scoring iterations: 4
```

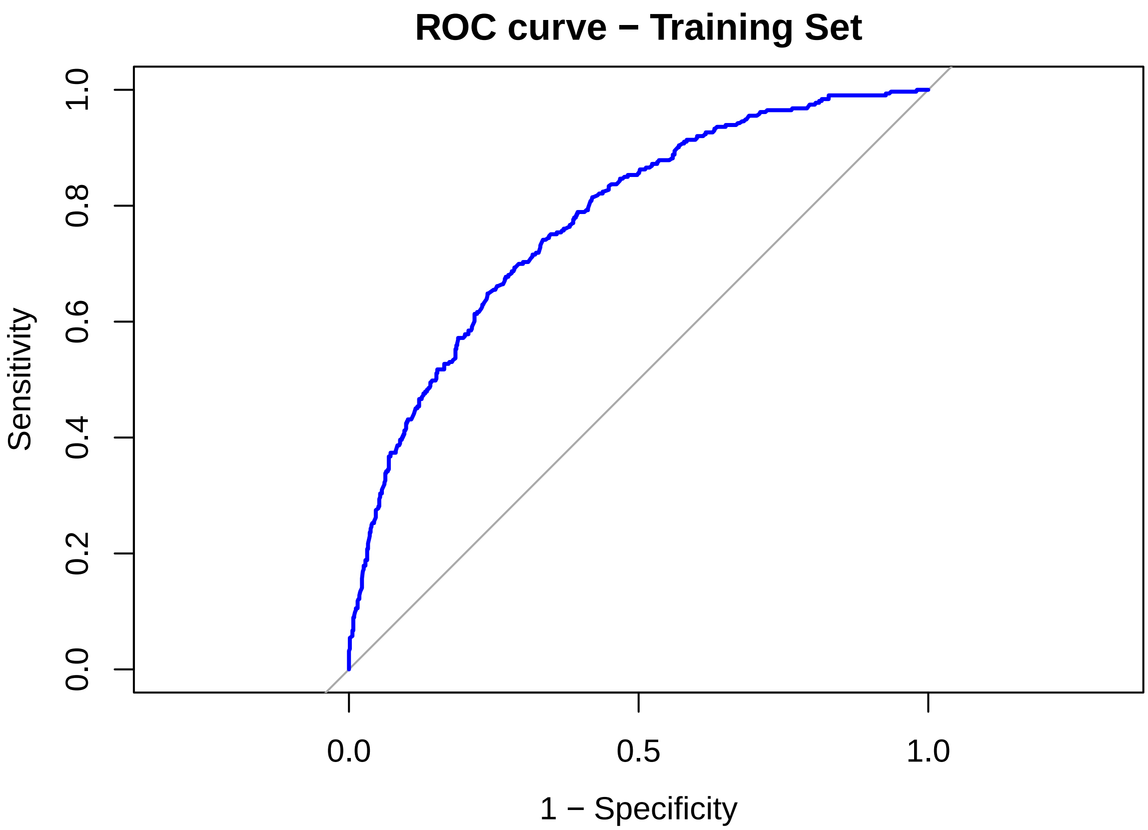
```
##
##                    0   1 Sum
##   (0,0.0756]      96   3  99
##   (0.0756,0.108]  90   8  98
##   (0.108,0.159]   83  16  99
##   (0.159,0.217]   78  19  97
##   (0.217,0.267]   72  29 101
##   (0.267,0.344]   69  26  95
##   (0.344,0.406]   58  40  98
##   (0.406,0.53]    57  42  99
##   (0.53,0.649]    41  56  97
##   (0.649,1]       25  74  99
##   Sum            669 313 982
```



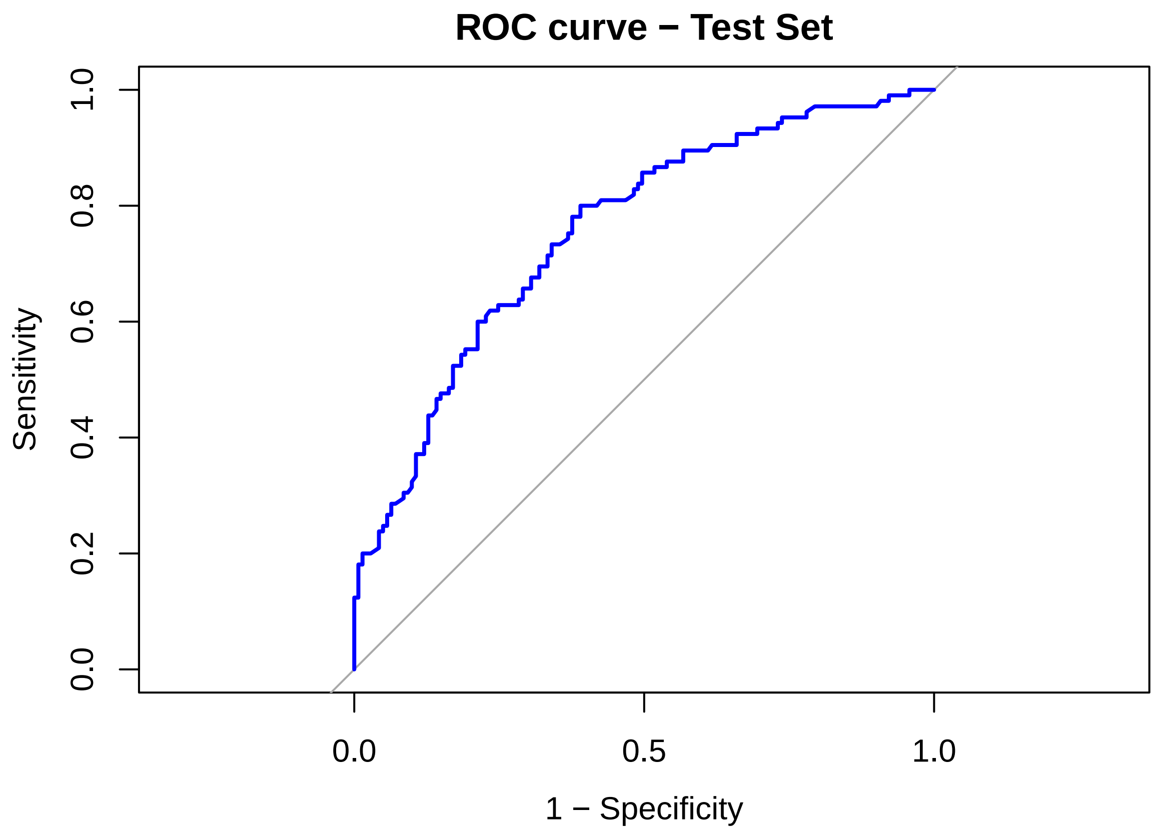

```
##
##                      0   1 Sum
##   (0,0.0886]        23   3  26
##   (0.0886,0.112]    20   4  24
##   (0.112,0.137]     18   6  24
##   (0.137,0.18]      18   7  25
##   (0.18,0.228]      14  10  24
##   (0.228,0.313]     15  10  25
##   (0.313,0.405]     10  14  24
##   (0.405,0.535]      9  17  26
##   (0.535,0.623]     10  13  23
##   (0.623,1]          4  21  25
##   Sum              141 105 246
```




```
##
## Call:
## roc.formula(formula = ED.train$admission ~ ED.train$phat_clin)
##
## Data: ED.train$phat_clin in 669 controls (ED.train$admission 0) < 313 cases (ED.train$admission 1).
## Area under the curve: 0.7758
```

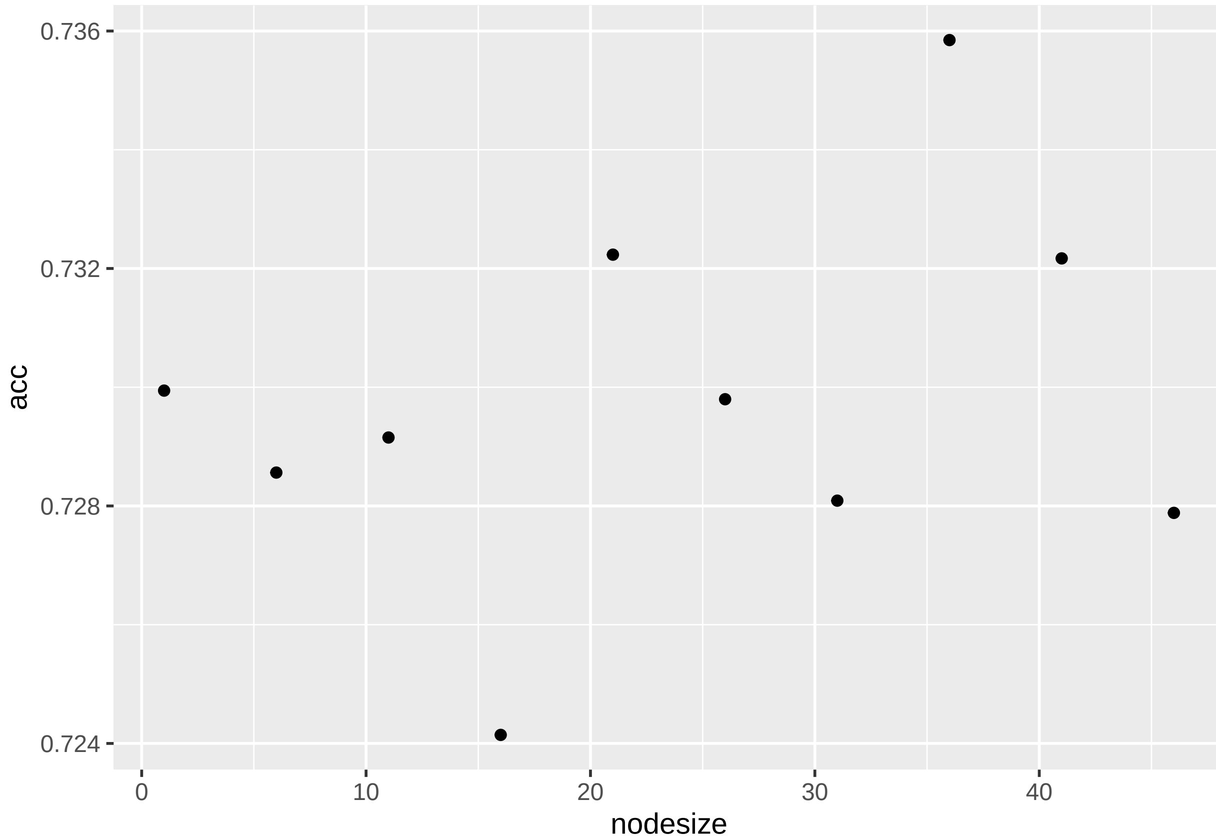



```
##
## Call:
## roc.formula(formula = ED.train$admission ~ ED.train$phat_clin)
##
## Data: ED.train$phat_clin in 669 controls (ED.train$admission 0) < 313 cases (ED.train$admission 1).
## Area under the curve: 0.7758
```




```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 105  39
##          1  36  66
##
##                Accuracy : 0.6951
##                  95% CI : (0.6335, 0.752)
##     No Information Rate : 0.5732
##     P-Value [Acc > NIR] : 5.542e-05
##
##                   Kappa : 0.3746
##
##  Mcnemar's Test P-Value : 0.8174
##
##             Sensitivity : 0.6286
##             Specificity : 0.7447
##          Pos Pred Value : 0.6471
##          Neg Pred Value : 0.7292
##              Prevalence : 0.4268
##          Detection Rate : 0.2683
##    Detection Prevalence : 0.4146
##       Balanced Accuracy : 0.6866
##
##        'Positive' Class : 1
```

```
##
```



Conclusion

Summary of your question, methods and results Additional topics can include: Was your analysis successful? Why or why not? What would you do if you had more time?

References

Data source: https://www.kaggle.com/datasets/ilkeryildiz/emergency-service-triage-application Original analysis: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0216972

Appendix

```
#read in data
library(readr)
emergency <- read_delim("~/Library/Mobile
↪  Documents/com~apple~CloudDocs/Fall2/BST260/emergency.csv",
   delim = ";", escape_double = FALSE, trim_ws = TRUE)
```

```
## Rows: 1267 Columns: 24
## -- Column specification --------------------------------------------------------
## Delimiter: ";"
## chr  (9): Chief_complain, NRS_pain, SBP, DBP, HR, RR, BT, Saturation, Diagno...
## dbl (14): Group, Sex, Age, Patients number per hour, Arrival mode, Injury, M...
## num  (1): KTAS duration_min
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

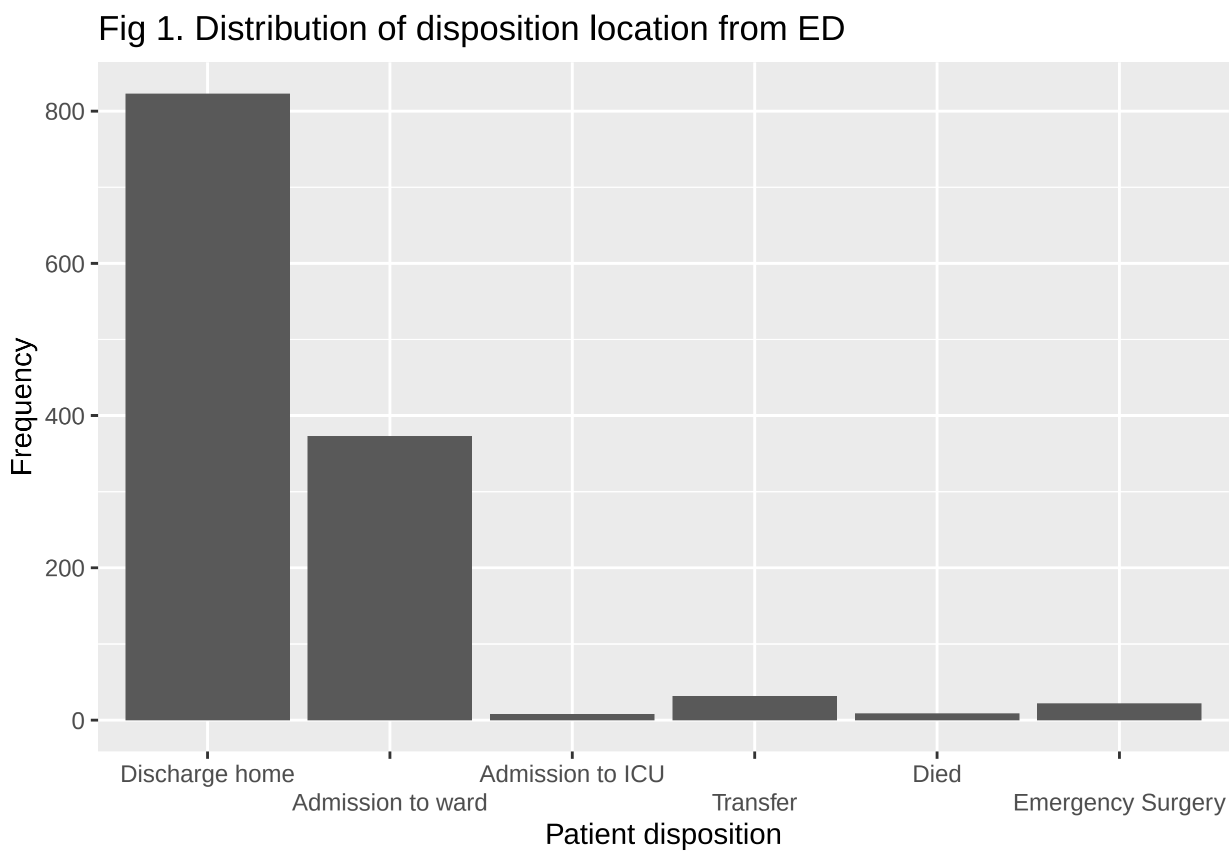
```r
#data exploration
library(dplyr)
library(ggplot2)
# outcome
## Disposition: 1 = Discharge, 2 = Admission to ward, 3 = Admission to ICU, 4 =
↪  Discharge, 5 = Transfer, 6 = Death, 7 = Surgery
summary(as.factor(emergency$Disposition))
```

```
##   1   2   3   4   5   6   7
## 797 373   8  26  32   9  22
```

```r
emergency$Disposition <- as.factor(ifelse(emergency$Disposition == 1 |
↪  emergency$Disposition == 4, 1, emergency$Disposition))
summary(emergency$Disposition)
```

```
##   1   2   3   5   6   7
## 823 373   8  32   9  22
```

```r
#hist and provide histogram
label <- c("Discharge home", "Admission to ward", "Admission to ICU", "Transfer", "Died",
↪  "Emergency Surgery")
emergency$disposition <- factor(emergency$Disposition, levels = c(1, 2, 3, 5, 6, 7),
↪  labels = label)
emergency |> ggplot() + geom_bar(aes(disposition)) + xlab("Patient disposition") +
↪  ylab("Frequency") + ggtitle("Fig 1. Distribution of disposition location from ED") +
↪  scale_x_discrete(guide = guide_axis(n.dodge=2))
```




```r
# binary outcome: discharge - disposition 1 of 4
emergency$admission <- ifelse(emergency$Disposition == 1 | emergency$Disposition == 4, 0,
↪  1)
summary(as.factor(emergency$admission))
```

```
##   0   1
## 823 444
```

```r
##data cleaning

# use as is
## sex: 1 female, 2 male
## age: continuous in years
## mental: 1 = Alert, 2 = Verbal Response, 3 = Pain Response, 4 = Unresponsive
## chief complaint: text
## pain: yes=1, no = 0
## SBP: systolic blood pressure
## DBP: diastolic blood pressure
## HR: heart rate
## KTAS_RN: 1 = resuscitation, 2 = emergent, 3 = urgent, 4 = less urgent, 5 = non-urgent

# delete
## group, which ED - delete
## patients number per hour - unclear, also should not affect emergency surgery
## saturation - many missing, and available values range from 90 to 100, not very
↪  pathologic, no high predictive value to be expected
```

```r
emergency <- emergency |> select(-Group, -`Patients number per hour`, -Saturation,
↪  -KTAS_expert, -Error_group, -mistriage, -`KTAS duration_min`)

#clean/rename
emergency$sbp <- as.numeric(emergency$SBP)
```

```
## Warning: NAs introduced by coercion
```

```r
summary(emergency$sbp)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    50.0   114.0   130.0   133.6   150.0   275.0      25
```

```r
emergency$dbp <- as.numeric(emergency$DBP)
```

```
## Warning: NAs introduced by coercion
```

```r
summary(emergency$dbp)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   31.00   70.00   80.00   79.78   90.00  160.00      29
```

```r
emergency$hr <- as.numeric(emergency$HR)
```

```
## Warning: NAs introduced by coercion
```

```r
summary(emergency$hr)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   32.00   72.00   82.00   83.96   96.00  148.00      20
```

```r
emergency$resp <- as.numeric(emergency$RR)
```

```
## Warning: NAs introduced by coercion
```

```r
summary(emergency$resp)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   14.00   18.00   20.00   19.51   20.00   30.00      22
```

```r
emergency$temp <- as.numeric(emergency$BT)
```

```
## Warning: NAs introduced by coercion
```

```r
summary(emergency$temp)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   35.00   36.20   36.50   36.58   36.80   41.00      18
```

```r
emergency <- emergency |> select(-SBP, -DBP, -HR, -RR, -BT)

# recategorize
##arrival mode: 1 = Walking, 2 = Public Ambulance, 3 = Private Vehicle, 4 = Private
↪  Ambulance, 5,6,7 = Other]
## -> 1 = walking, 2 = ambulance, 3 = private vehicle, 4 = other
emergency$arrival <- ifelse(emergency$`Arrival mode`==2 | emergency$`Arrival mode`==4, 2,
↪  emergency$`Arrival mode`)
emergency$arrival <- ifelse(emergency$arrival==5 | emergency$arrival==6 |
↪  emergency$arrival==7, 4, emergency$arrival)
summary(as.factor(emergency$arrival))
```

```
##   1   2   3   4
##  79 421 753  14
```

```r
## injury: 2=yes, 1=no -> 1 yes, 0 no
emergency$injury <- ifelse(emergency$Injury==2, 1, 0)
summary(as.factor(emergency$injury))
```

```
##    0    1
## 1023  244
```

```r
emergency <- emergency |> select(-Injury, -`Arrival mode`)


##NRS_pain: replace missing as 0, if they did not have pain
emergency$NRS_pain <-  as.numeric(emergency$NRS_pain)
```

```
## Warning: NAs introduced by coercion
```

```r
emergency$NRS_pain <-  ifelse(is.na(emergency$NRS_pain) & emergency$Pain==0, 0,
↪  emergency$NRS_pain)
summary(as.factor(emergency$NRS_pain))
```

```
##    0    1    2    3    4    5    6    7    8    9   10 NA's
##  553    2   38  278  141  136   70   33    9    1    3    3
```

```r
# generate additional out of existing predictors:
## shock index: HR/SBP
emergency$shock_index <- emergency$hr/emergency$sbp
summary(emergency$shock_index)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##  0.2286  0.5221  0.6375  0.6525  0.7455  1.5934      25
```

```r
## shock: shock index > 1
emergency$shock <- ifelse(emergency$shock_index > 1, 1, 0)
summary(as.factor(emergency$shock))
```

```
##    0    1 NA's
## 1197   45   25
```

```r
## hyperventilation: respiratory rate > 25
emergency$hyperventilation <- ifelse(emergency$resp > 25, 1, 0)
summary(as.factor(emergency$shock))
```

```
##    0    1 NA's
## 1197   45   25
```

```r
#fix some column_names
emergency$ED_diagnosis <- emergency$`Diagnosis in ED`
emergency$ED_LOS_min <- emergency$`Length of stay_min`
emergency$chief_complaint <- emergency$Chief_complain
emergency$mental_status <- emergency$Mental
emergency$pain_yn <- emergency$Pain

## fever based on body temperature?
emergency$hypertherm <- ifelse(emergency$temp > 37.5 & !is.na(emergency$temp), 1, 0)
emergency <- emergency |> select(-`Diagnosis in ED`, -`Length of stay_min`,
↪  -Chief_complain, -Mental, -Pain)

## create a complete case cohort -> drop anyone with any missings as all variables kept
↪  in the dataset will be used for machine learning approach
ED_complete <- na.omit(emergency)
#creating a validation set with 20% of data
smp_size <- floor(0.80 * nrow(ED_complete))

## set the seed
set.seed(2404)
train_ind <- sample(seq_len(nrow(ED_complete)), size = smp_size)

ED.train <- ED_complete[train_ind, ]
ED.test <- ED_complete[-train_ind, ]

#create table with inpatient admission rate for the whole dataset, training and
↪  validation model
# check outcome in separated sets
addmargins(table(as.factor(ED_complete$admission)));
↪  prop.table(table(as.factor(ED_complete$admission)))
```

```
##
##    0    1  Sum
##  810  418 1228
```

```
##
##         0         1
## 0.6596091 0.3403909
```

```r
# full dataset: patients with hearing difficulty: 1341/8798 (15.24%)
addmargins(table(as.factor(ED.train$admission)));
↪  prop.table(table(as.factor(ED.train$admission)))
```

```
##
##   0   1 Sum
## 669 313 982

##
##         0         1
## 0.6812627 0.3187373
```

```r
# training sample: patients with hearing difficulty: 1007/6598 (15.26%)


#training the clinician model: sex, age, NRS_pain, KTAS_RN, arrival, injury, shock,
↪  hyperventilation, mental status, hyperthermia/fever
library(caret)
#create a table displaying characteristics by outcome including the preselected variables

#stepwise forward regression with AIC as criterion
library(MASS)
```

```
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
```

```r
Fitall.tr <- glm(admission ~ as.factor(Sex) + Age + NRS_pain + KTAS_RN +
↪  as.factor(arrival) + injury + shock + hyperventilation + mental_status + hypertherm,
↪  family="binomial", data= ED.train)

Fitstart <- glm(admission ~ 1, family="binomial", data= ED.train)

set.seed(2024)
m_clin <- stepAIC(Fitstart, scope=formula(Fitall.tr), direction="forward", k=2)
```

```
## Start:  AIC=1231.3
## admission ~ 1
##
##                      Df Deviance    AIC
## + KTAS_RN             1   1093.9 1097.9
## + as.factor(arrival)  3   1147.3 1155.3
## + Age                 1   1177.8 1181.8
## + injury              1   1197.2 1201.2
## + hypertherm          1   1218.0 1222.0
## + shock               1   1220.7 1224.7
## + NRS_pain            1   1221.7 1225.7
```

```
## + hyperventilation    1   1224.4 1228.4
## + mental_status       1   1226.4 1230.4
## + as.factor(Sex)      1   1226.7 1230.7
## <none>                    1229.3 1231.3
##
## Step:  AIC=1097.88
## admission ~ KTAS_RN
##
##                      Df Deviance    AIC
## + Age                 1   1064.2 1070.2
## + as.factor(arrival)  3   1062.3 1072.3
## + injury              1   1081.4 1087.4
## + hypertherm          1   1083.4 1089.4
## + shock               1   1090.1 1096.1
## + NRS_pain            1   1090.6 1096.6
## <none>                    1093.9 1097.9
## + as.factor(Sex)      1   1092.7 1098.7
## + hyperventilation    1   1093.2 1099.2
## + mental_status       1   1093.7 1099.7
##
## Step:  AIC=1070.2
## admission ~ KTAS_RN + Age
##
##                      Df Deviance    AIC
## + as.factor(arrival)  3   1040.0 1052.0
## + hypertherm          1   1051.9 1059.9
## + injury              1   1056.3 1064.3
## + shock               1   1059.5 1067.5
## + as.factor(Sex)      1   1062.2 1070.2
## <none>                    1064.2 1070.2
## + NRS_pain            1   1063.3 1071.3
## + mental_status       1   1063.7 1071.7
## + hyperventilation    1   1063.7 1071.7
##
## Step:  AIC=1052.02
## admission ~ KTAS_RN + Age + as.factor(arrival)
##
##                    Df Deviance    AIC
## + hypertherm        1   1027.9 1041.9
## + injury            1   1029.7 1043.7
## + shock             1   1034.6 1048.6
## + mental_status     1   1037.6 1051.6
## <none>                  1040.0 1052.0
## + as.factor(Sex)    1   1039.2 1053.2
## + NRS_pain          1   1039.5 1053.5
## + hyperventilation  1   1039.7 1053.7
##
## Step:  AIC=1041.89
## admission ~ KTAS_RN + Age + as.factor(arrival) + hypertherm
##
##                    Df Deviance    AIC
## + injury            1   1019.5 1035.5
## + shock             1   1023.0 1039.0
## + mental_status     1   1024.4 1040.4
```

```
## <none>                   1027.9 1041.9
## + as.factor(Sex)    1    1027.2 1043.2
## + NRS_pain          1    1027.6 1043.6
## + hyperventilation  1    1027.9 1043.9
##
## Step:  AIC=1035.55
## admission ~ KTAS_RN + Age + as.factor(arrival) + hypertherm +
##     injury
##
##                    Df Deviance    AIC
## + shock             1   1014.8 1032.8
## + mental_status     1   1016.7 1034.7
## <none>                  1019.5 1035.5
## + as.factor(Sex)    1   1018.4 1036.4
## + NRS_pain          1   1019.5 1037.5
## + hyperventilation  1   1019.5 1037.5
##
## Step:  AIC=1032.83
## admission ~ KTAS_RN + Age + as.factor(arrival) + hypertherm +
##     injury + shock
##
##                    Df Deviance    AIC
## + mental_status     1   1011.6 1031.7
## <none>                  1014.8 1032.8
## + as.factor(Sex)    1   1013.6 1033.6
## + hyperventilation  1   1014.8 1034.8
## + NRS_pain          1   1014.8 1034.8
##
## Step:  AIC=1031.65
## admission ~ KTAS_RN + Age + as.factor(arrival) + hypertherm +
##     injury + shock + mental_status
##
##                    Df Deviance    AIC
## <none>                  1011.6 1031.7
## + as.factor(Sex)    1   1010.2 1032.2
## + NRS_pain          1   1011.6 1033.6
## + hyperventilation  1   1011.6 1033.7
```

```
summary(m_clin)
```

```
##
## Call:
## glm(formula = admission ~ KTAS_RN + Age + as.factor(arrival) +
##     hypertherm + injury + shock + mental_status, family = "binomial",
##     data = ED.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8111  -0.7988  -0.4736   0.9040   2.4661
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)           0.374504   0.622920   0.601 0.547702
```

```
## KTAS_RN             -0.790070   0.103579  -7.628 2.39e-14 ***
## Age                  0.019035   0.004234   4.496 6.93e-06 ***
## as.factor(arrival)2  1.362495   0.400187   3.405 0.000663 ***
## as.factor(arrival)3  0.572665   0.391449   1.463 0.143484
## as.factor(arrival)4 -0.258425   0.879563  -0.294 0.768903
## hypertherm           1.100834   0.337552   3.261 0.001109 **
## injury              -0.643295   0.242522  -2.653 0.007989 **
## shock                0.875347   0.388884   2.251 0.024391 *
## mental_status       -0.426530   0.239006  -1.785 0.074326 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1229.3  on 981  degrees of freedom
## Residual deviance: 1011.6  on 972  degrees of freedom
## AIC: 1031.6
##
## Number of Fisher Scoring iterations: 4
```

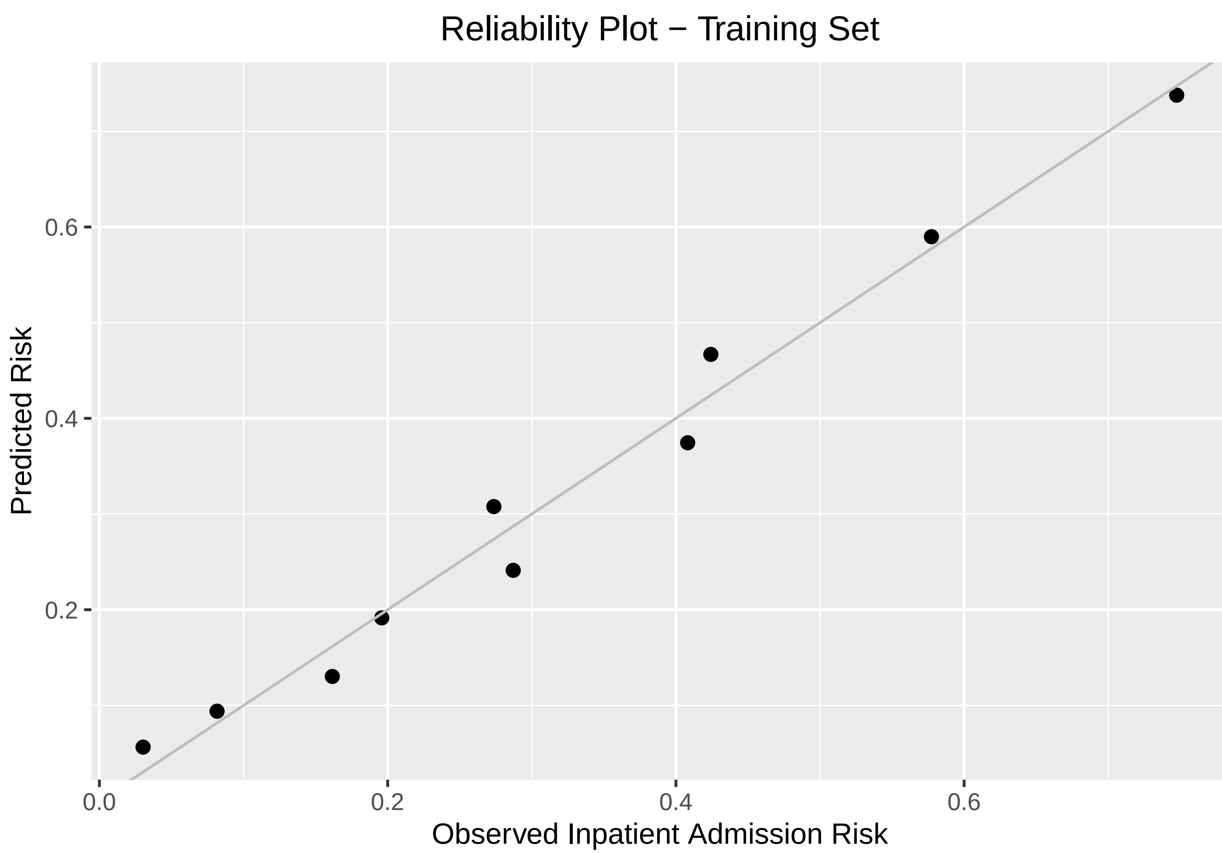
```
#aaply to validation set
ED.train$phat_clin <- predict(m_clin, type="response", newdata=ED.train)
ED.test$phat_clin <- predict(m_clin, type="response", newdata=ED.test)

#Calibration Plot -training set
##create risk deciles on predicted risk
cuts <- quantile(ED.train$phat_clin, prob=c(.1,.2,.3,.4,.5,.6,.7,.8,.9), na.rm=T)
ED.train$risk_decile <-cut(ED.train$phat_clin, breaks=c(0, cuts, 1))
dec<-c(1:10) #for plot
#observed proportion of difficult hearing in risk deciles
t1.train<-table(ED.train$risk_decile, ED.train$admission)
addmargins(t1.train)
```

```
##
##                    0   1 Sum
##   (0,0.0756]      96   3  99
##   (0.0756,0.108]  90   8  98
##   (0.108,0.159]   83  16  99
##   (0.159,0.217]   78  19  97
##   (0.217,0.267]   72  29 101
##   (0.267,0.344]   69  26  95
##   (0.344,0.406]   58  40  98
##   (0.406,0.53]    57  42  99
##   (0.53,0.649]    41  56  97
##   (0.649,1]       25  74  99
##   Sum            669 313 982
```

```
t2.train <- prop.table(t1.train, 1)
obs.train <- t2.train[,2] #for plot
#mean predicted risk in risk deciles
deciles.train <- ED.train %>% group_by(risk_decile) %>% summarise(mean=mean(phat_clin))
pred.train <- deciles.train$mean #for plot
```

```
cali_train<-data.frame(dec, obs.train, pred.train) # for plot
ggplot(cali_train, aes(x=obs.train, y=pred.train)) + geom_point(size=2) + xlab("Observed
↪  Inpatient Admission Risk") + ylab("Predicted Risk") + ggtitle("Reliability Plot -
↪  Training Set") + theme(plot.title = element_text(hjust = 0.5)) +
↪  geom_abline(intercept = 0, slope = 1, color="grey")
```

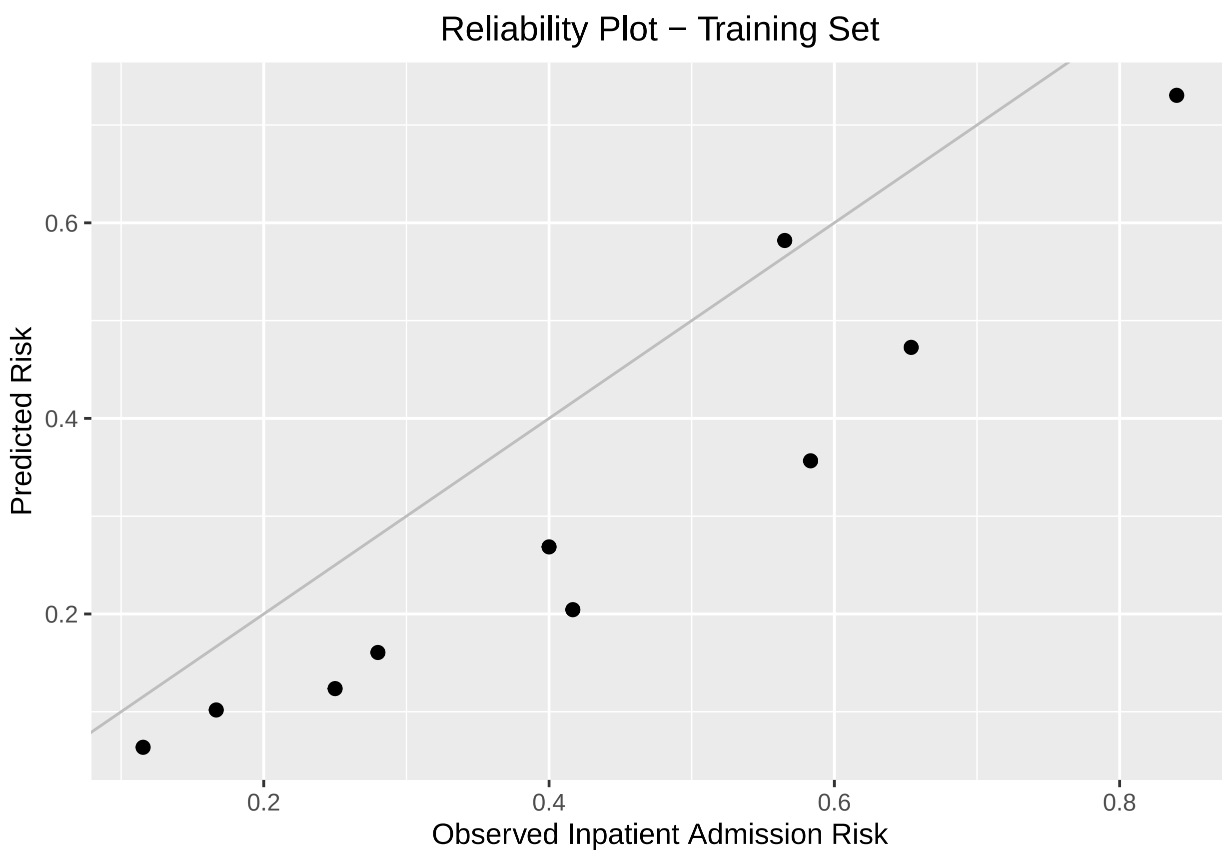



```
#Calibration Plot -validation set
##create risk deciles on predicted risk
cuts <- quantile(ED.test$phat_clin, prob=c(.1,.2,.3,.4,.5,.6,.7,.8,.9), na.rm=T)
ED.test$risk_decile <-cut(ED.test$phat_clin, breaks=c(0, cuts, 1))
dec<-c(1:10) #for plot
#observed proportion of difficult hearing in risk deciles
t1.test<-table(ED.test$risk_decile, ED.test$admission)
addmargins(t1.test)
```

```
##
##                      0   1 Sum
##   (0,0.0886]        23   3  26
##   (0.0886,0.112]    20   4  24
##   (0.112,0.137]     18   6  24
##   (0.137,0.18]      18   7  25
##   (0.18,0.228]      14  10  24
##   (0.228,0.313]     15  10  25
##   (0.313,0.405]     10  14  24
```

```
##    (0.405,0.535]    9  17  26
##    (0.535,0.623]   10  13  23
##    (0.623,1]        4  21  25
##    Sum            141 105 246
```

```
t2.test <- prop.table(t1.test, 1)
obs.test <- t2.test[,2] #for plot
#mean predicted risk in risk deciles
deciles.test <- ED.test %>% group_by(risk_decile) %>% summarise(mean=mean(phat_clin))
pred.test <- deciles.test$mean #for plot
cali_test<-data.frame(dec, obs.test, pred.test) # for plot
ggplot(cali_test, aes(x=obs.test, y=pred.test)) + geom_point(size=2) + xlab("Observed
↪  Inpatient Admission Risk") + ylab("Predicted Risk") + ggtitle("Reliability Plot -
↪  Training Set") + theme(plot.title = element_text(hjust = 0.5)) +
↪  geom_abline(intercept = 0, slope = 1, color="grey")
```



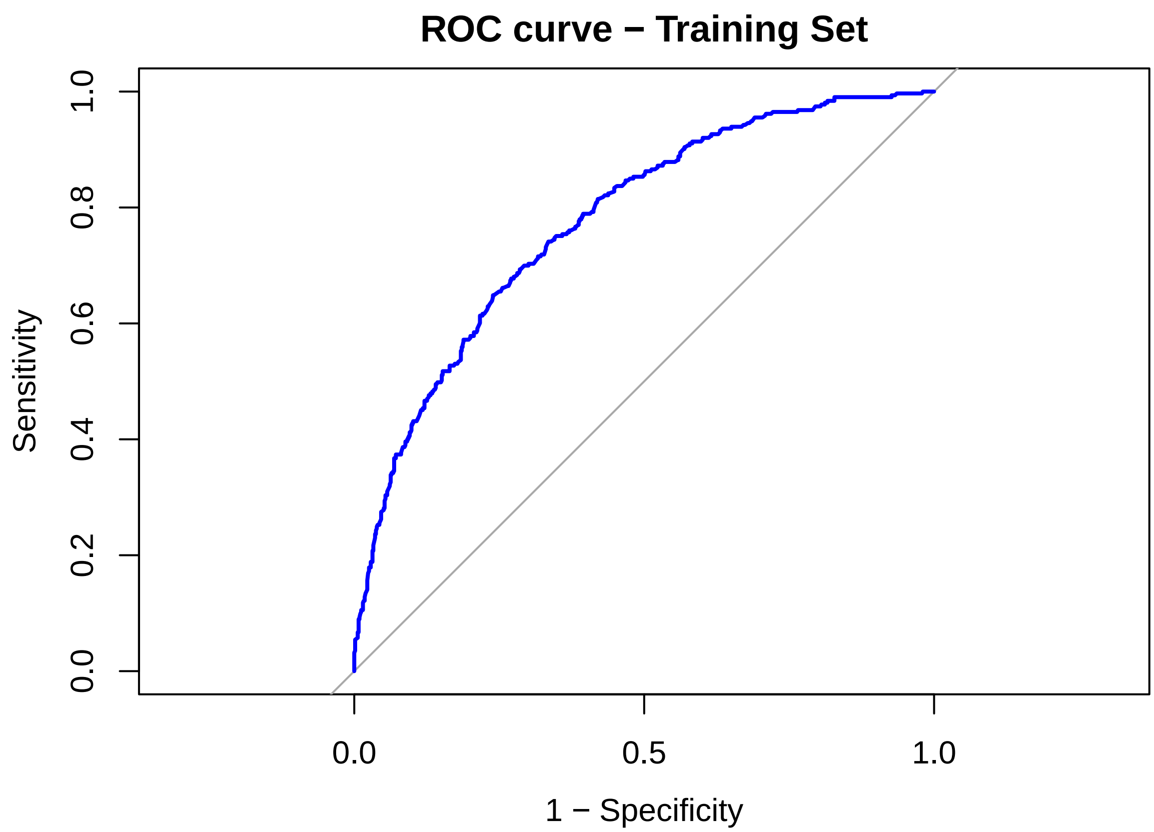

```
#AUC to find good cutoff for predicting binary outcome based on predicted risk
library(pROC)
roccurve.train<- roc(ED.train$admission ~ ED.train$phat_clin); roccurve.train
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
##
```

```
## Call:
## roc.formula(formula = ED.train$admission ~ ED.train$phat_clin)
##
## Data: ED.train$phat_clin in 669 controls (ED.train$admission 0) < 313 cases (ED.train$admission 1).
## Area under the curve: 0.7758
```

```
plot(roccurve.train, legacy.axes=T, main="ROC curve - Training Set", col="blue")
```

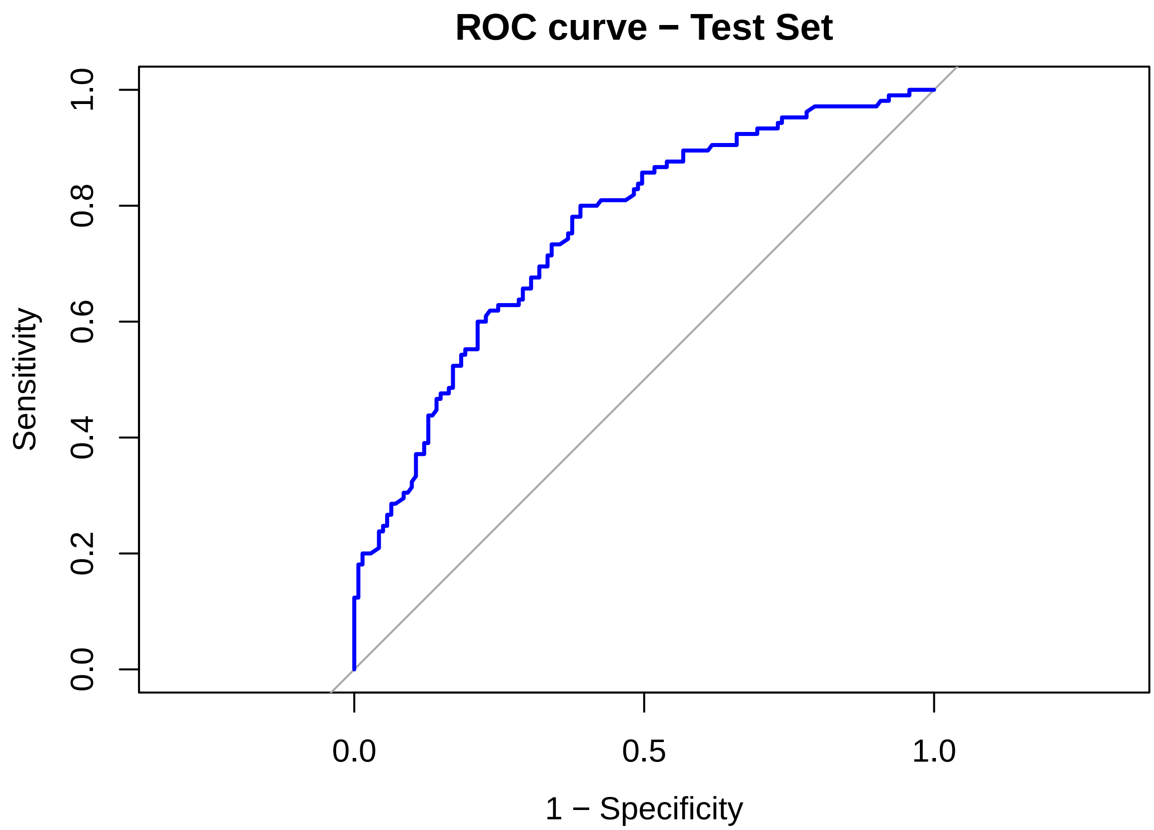



```
roccurve.test <- roc(ED.test$admission ~ ED.test$phat_clin); roccurve.train
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
##
## Call:
## roc.formula(formula = ED.train$admission ~ ED.train$phat_clin)
##
## Data: ED.train$phat_clin in 669 controls (ED.train$admission 0) < 313 cases (ED.train$admission 1).
## Area under the curve: 0.7758
```

```
plot(roccurve.test, legacy.axes=T, main="ROC curve - Test Set", col="blue")
```




```
#check sensitivity and specificity for different cutoffs of predicted risk in training
↪  set
ED.train$admission <- as.factor(ED.train$admission)
## 0.3
ED.train$yhat_clin03 <- as.factor(ifelse(ED.train$phat_clin > 0.3, 1, 0))
cm_clinical_train03 <- confusionMatrix(ED.train$yhat_clin03, ED.train$admission,
↪  positive="1")
## 0.4
ED.train$yhat_clin04 <- as.factor(ifelse(ED.train$phat_clin > 0.4, 1, 0))
cm_clinical_train04 <- confusionMatrix(ED.train$yhat_clin04, ED.train$admission,
↪  positive="1")
## 0.5
ED.train$yhat_clin05 <- as.factor(ifelse(ED.train$phat_clin > 0.5, 1, 0))
cm_clinical_train05 <- confusionMatrix(ED.train$yhat_clin05, ED.train$admission,
↪  positive="1")
##0.6
ED.train$yhat_clin06 <- as.factor(ifelse(ED.train$phat_clin > 0.6, 1, 0))
cm_clinical_train06 <- confusionMatrix(ED.train$yhat_clin06, ED.train$admission,
↪  positive="1")
##0.7
ED.train$yhat_clin07 <- as.factor(ifelse(ED.train$phat_clin > 0.7, 1, 0))
cm_clinical_train07 <- confusionMatrix(ED.train$yhat_clin07, ED.train$admission,
↪  positive="1")
##0.8
ED.train$yhat_clin08 <- as.factor(ifelse(ED.train$phat_clin > 0.8, 1, 0))
cm_clinical_train08 <- confusionMatrix(ED.train$yhat_clin08, ED.train$admission,
↪  positive="1")
```

```r
# best trade-off between sensitvity and secificity: cutoff: 0.3

#performance parameters for different cutoffs
rownames <- c("0.3","0.4","0.5", "0.6", "0.7", "0.8")
Specificity <-
↪  c(cm_clinical_train03$byClass["Specificity"],cm_clinical_train04$byClass["Specificity"],cm_clinical_
Sensitivity <-
↪  c(cm_clinical_train03$byClass["Sensitivity"],cm_clinical_train04$byClass["Sensitivity"],cm_clinical_
Accuracy <- c(cm_clinical_train03$overall["Accuracy"],
↪  cm_clinical_train04$overall["Accuracy"],cm_clinical_train05$overall["Accuracy"],
↪  cm_clinical_train06$overall["Accuracy"], cm_clinical_train07$overall["Accuracy"],
↪  cm_clinical_train08$overall["Accuracy"])
Table_cutoff <- data.frame(row.names=rownames, Sensitivity, Specificity, Accuracy)
# best trade-off between sensitvity and secificity: cutoff: 0.3

#Accuracy in test set

ED.test$yhat_clin03 <- as.factor(ifelse(ED.test$phat_clin > 0.3, 1, 0))
ED.test$admission <- as.factor(ED.test$admission)
cm_clinical_test <- confusionMatrix(ED.test$yhat_clin03, ED.test$admission, positive="1")
cm_clinical_test
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 105  39
##          1  36  66
##
##                Accuracy : 0.6951
##                  95% CI : (0.6335, 0.752)
##     No Information Rate : 0.5732
##     P-Value [Acc > NIR] : 5.542e-05
##
##                   Kappa : 0.3746
##
##  Mcnemar's Test P-Value : 0.8174
##
##             Sensitivity : 0.6286
##             Specificity : 0.7447
##          Pos Pred Value : 0.6471
##          Neg Pred Value : 0.7292
##              Prevalence : 0.4268
##          Detection Rate : 0.2683
##    Detection Prevalence : 0.4146
##       Balanced Accuracy : 0.6866
##
##        'Positive' Class : 1
##
```

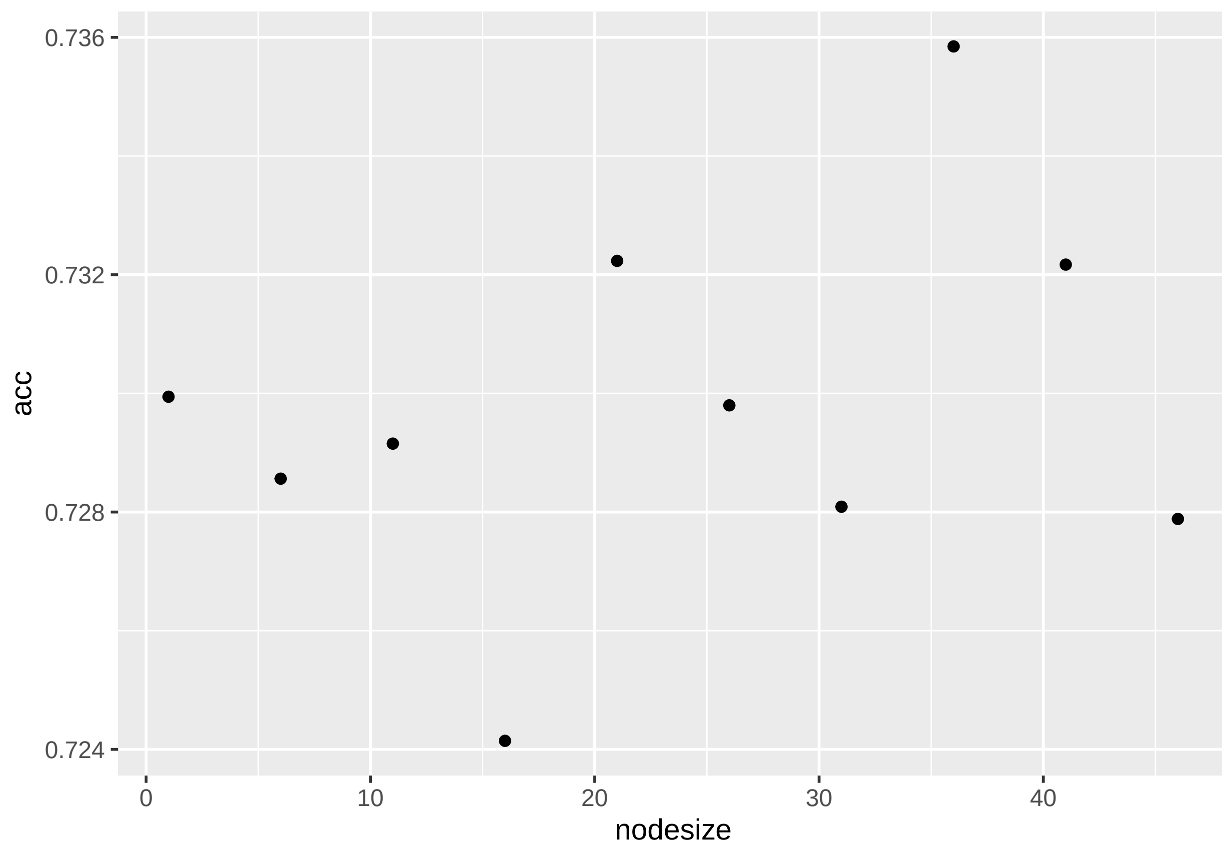
```r
#random forest
detach("package:MASS", unload = TRUE)
```

```
## Warning: 'MASS' namespace cannot be unloaded:
##   namespace 'MASS' is imported by 'ipred' so cannot be unloaded
```

```r
library(randomForest)

y_train <- ED.train$admission
x_train <- ED.train |> select(Sex, Age, NRS_pain, KTAS_RN, sbp, dbp, hr, resp, temp,
↪  arrival, injury, shock_index, mental_status)

#tuning nodesize
set.seed(2404)
nodesize <- seq(1, 50, 5)
acc <- sapply(nodesize, function(ns){
  train(data.frame(x_train), factor(y_train), method = "rf",
               tuneGrid = data.frame(mtry = 5),
               nodesize = ns)$results$Accuracy
})
qplot(nodesize, acc)
```



```r
#fit random forest model
set.seed(2333)
fit_rf <- randomForest(data.frame(x_train), factor(y_train),
                       mtry = 5, nodesize = nodesize[which.max(acc)])

#random forest model performance in internal validation set
set.seed(2134)
ED.train$yhat_ML <- predict(fit_rf, type="response", newdata=ED.train)
ED.test$yhat_ML <- predict(fit_rf, type="response", newdata=ED.test)

#Accuracy in test set
#cm_ML_forest_test <- confusionMatrix(factor(ED.test$yhat_ML), factor(ED.test$admission),
↪  positive="1")
#cm_ML_forest_test
```

$$\hat{p} = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 x}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 x}}$$

For the intercept only logistic model: $\hat{\beta}_0 = -3.33$ and therefore the calculated probability of having diabetes: $\hat{p} = \frac{e^{\hat{\beta}_0}}{1+e^{\hat{\beta}_0}} = 0.034489568$

177/5132 = 0.03448948 (3.45%) - yes, this equals the calculated probability based on the intercept only model

Model	Sensitivity	FPF
Simple (5% FPF)	0.1885	0.05
Clinical (5% FPF)	0.2377	0.05
Simple (3% FPF)	0.1148	0.03
Clinical (3% FPF)	0.1885	0.03