STAT 471: Midterm Exam Solutions

[Name]

March 22, 2020

Contents

Instructions

- This exam is open-book / open-notes / open-internet. However, it is individual work. Communication among students is prohibited.
- Please complete your homework in R Markdown, using this document as a starting point. Show your R code using code chunks and add your text answers using **bold text**.
- When you are ready to submit, please compile your R Markdown file into a PDF. Then, submit this PDF through Canvas.
- While base R programming is acceptable, I strongly encourage you to use the `tidyverse` to complete your assignment. However, points will not be deducted if you use base R programming.
- In addition, please make sure to have clear labels and sensible titles on your plots.
- **Make sure that `readmission_clean.csv` and `plot_glmnet.R` are in your working directory before beginning the exam.**

Introduction: Predicting readmission for diabetes patients

Background and goals

Diabetes is a chronic medical condition affecting millions of Americans, but if managed well, patients can lead relatively normal lives. However, if improperly managed, diabetes can lead to patients being continuously admitted and readmitted to hospitals. Hospital readmissions represent a failure of the health system to provide adequate support to the patient and are extremely costly to the system. The goals are therefore to

1. identify important factors associated with readmission, and
2. predict whether a given patient will be readmitted.

Readmission data

In this exam, we will investigate a dataset originally from the Center for Clinical and Translational Research at Virginia Commonwealth University, covering diabetes patients across 130 U.S. hospitals from 1999 to 2008. Three former STAT 471 students Spencer Luster, Matthew Lesser and Mridul Ganesh brought this data set into the class through their final project. In this exam, we will use a cleaned subset of this data.

First, let's load a few libraries:

```r
library(glmnet)
library(scales)
library(tidyverse)
```

Let's also source `plot_glmnet.R` to get access to the `plot_glmnet` function from Unit 3 Lecture 2 (make sure this file is in your working directory).

```r
source("plot_glmnet.R")
```

Finally, let's load the data:

```r
readmission = read_csv("readmission_clean.csv")
readmission
```

```
## # A tibble: 99,492 x 26
##    race  gender age_group num_outpatient num_inpatient num_emergency
##    <chr> <chr>  <chr>              <dbl>         <dbl>         <dbl>
##  1 Cauc~ Female 80+                    0             0             0
##  2 Cauc~ Female 80+                    0             0             0
##  3 Cauc~ Male   20-59                  0             0             0
##  4 Afri~ Female 20-59                  0             0             0
##  5 Cauc~ Male   20-59                  0             0             0
##  6 Afri~ Male   60-79                  0             0             0
##  7 Cauc~ Female 20-59                  0             0             1
##  8 Cauc~ Male   80+                    0             0             0
##  9 Cauc~ Male   60-79                  0             0             0
## 10 Afri~ Female 60-79                  0             0             0
## # ... with 99,482 more rows, and 20 more variables: num_medications <dbl>,
## #   num_diagnoses <dbl>, adm_source <chr>, adm_type <chr>,
## #   time_in_hospital <dbl>, num_lab_procedures <dbl>, num_procedures <dbl>,
## #   discharge <chr>, max_glu_serum <chr>, A1Cresult <chr>, med_changed <chr>,
## #   med_prescribed <chr>, insulin <chr>, metformin <chr>, glimepiride <chr>,
## #   glipizide <chr>, glyburide <chr>, pioglitazone <chr>, rosiglitazone <chr>,
## #   readmitted <dbl>
```

Each row corresponds to a hospital admission of a patient. There are 26 total variables, described below:

Demographic variables

- `race`: patient's race
- `gender`: patient's gender
- `age_group`: patient's age group

Medical history

- `num_outpatient`: number of outpatient visits by the patient in the year prior to the current admission
- `num_inpatient`: number of inpatient visits by the patient in the year prior to the current admission
- `num_emergency`: number of emergency visits by the patient in the year prior to the current admission
- `num_medications`: number of total medications the patient has taken
- `num_diagnoses`: number of total diagnoses the patient has

Hospital admission details

- adm_source: who referred the patient to the hospital
- adm_type: type of admission
- time_in_hospital: length of stay in the hospital (in days)
- num_lab_procedures: number of lab procedures performed
- num_procedures: number of non-lab procedures performed
- discharge: where the patient was discharged

Clinical results

- max_glu_serum: results of glucose serum test
- A1Cresult: results of A1c test

Medication details

- med_changed: whether any medication was changed
- med_prescribed: whether any medication was prescribed
- insulin: type of change (if any) to insulin medication
- metformin: type of change (if any) to insulin medication
- glimepiride: type of change (if any) to glimepiride medication
- glipizide: type of change (if any) to glipizide medication
- glyburide: type of change (if any) to glyburide medication
- pioglitazone: type of change (if any) to pioglitazone medication
- rosiglitazone: type of change (if any) to rosiglitazone medication

Readmission indicator

- readmitted: whether the patient was readmitted to the hospital within 30 days of discharge

Train/test split

Let's split off 5000 observations for training and leave the rest for testing:

```r
set.seed(1)
n_total = nrow(readmission)
n_train = 5000
n_test = n_total-n_train
partition = sample(c(rep("train", n_train), rep("test", n_test)))
readmission_train = readmission %>%
  bind_cols(partition = partition) %>%
  filter(partition == "train") %>%
  select(-partition)
readmission_test = readmission %>%
  bind_cols(partition = partition) %>%
  filter(partition == "test") %>%
  select(-partition)
```

1 Exploratory data analysis

First, let's do some exploratory data analysis on our training data readmission_train.

1. What fraction of the patients in the training data were readmitted?

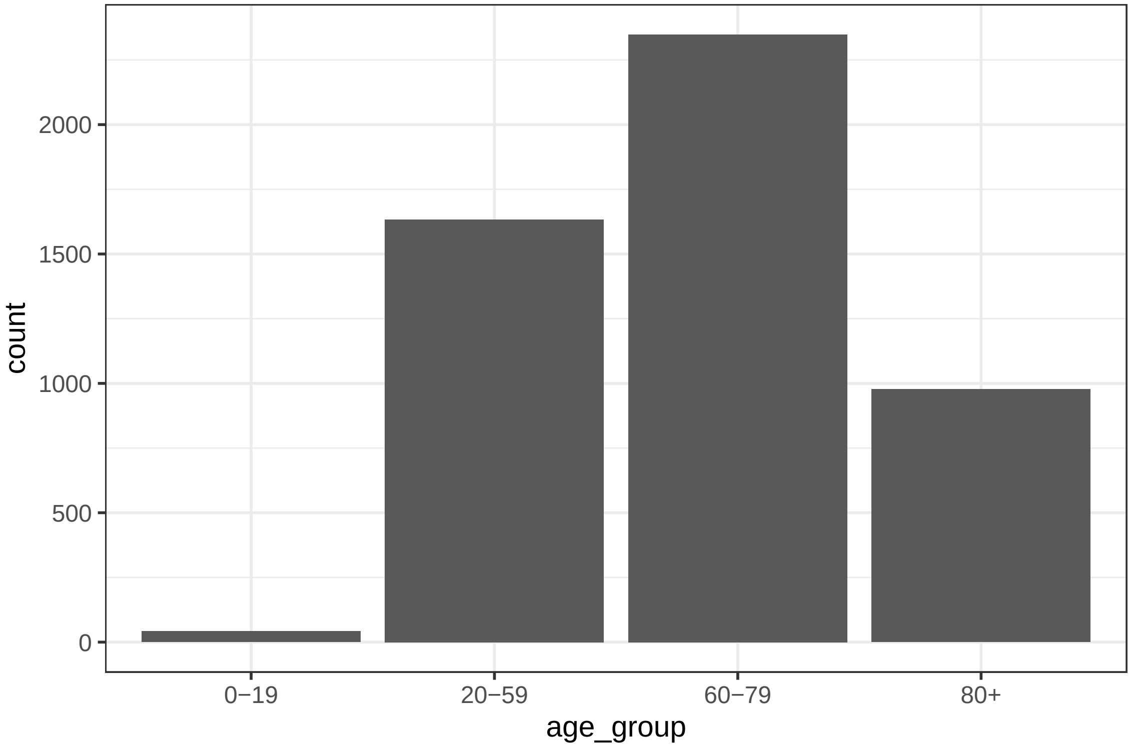
```r
readmission_train %>% summarise(frac_readmitted = mean(readmitted))
```

```
## # A tibble: 1 x 1
##   frac_readmitted
##              <dbl>
## 1            0.117
```

We see that 0.117 of the patients in the training data were readmitted.

2. Produce a bar plot to display the breakdown of the patients by age group. What is the most prevalent age group in the training data?

```
readmission_train %>%
  ggplot(aes(x = age_group)) + geom_bar() + theme_bw()
```

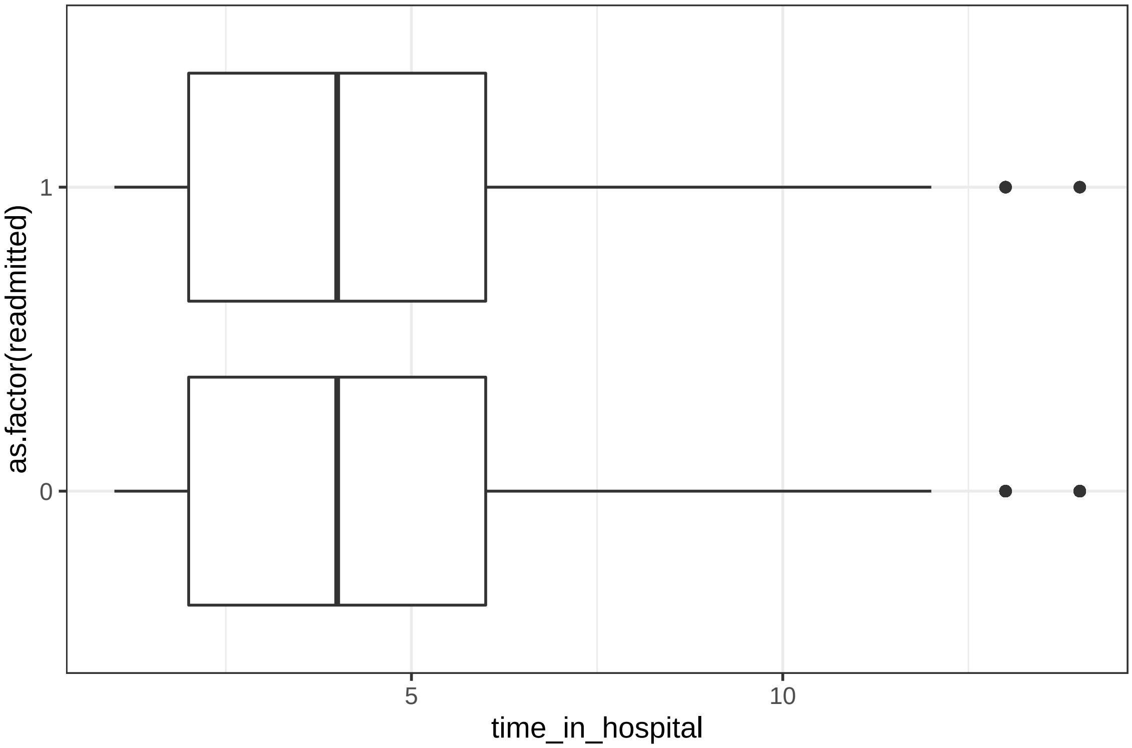


The most prevalent age group is 60-79.

3. Produce a plot to show the relationship between `time_in_hospital` and `readmitted`. Using `summarise`, compute the median time in hospital separately for patients that were not readmitted and for those that were. Do these suggest that readmission rates vary based on time in hospital, and if so, what is the direction of the relationship?

(Hint: It may be useful to convert `readmitted` to a factor using `as.factor(readmitted)`.)

```
readmission_train %>% ggplot(aes(x = time_in_hospital, y = as.factor(readmitted))) +
  geom_boxplot() + theme_bw()
```



```
readmission_train %>%
  group_by(readmitted) %>%
  summarise(median_time_in_hospital = median(time_in_hospital))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 2
##   readmitted median_time_in_hospital
##        <dbl>                   <dbl>
## 1          0                       4
## 2          1                       4
```

The median number of days in the hospital was 4 days for both groups of patients. This fact, along with the plot above, suggests that time in hospital is not associated with readmission.

2 Association via logistic regression

Next, let's explore the factors associated with hospital readmission using logistic regression on the training data.

4. Run a logistic regression of `readmitted` on all of the features and print the summary. Based on this summary, how many features are significantly associated with `readmitted` at the 0.05 level? Which of these are also significant at the 0.01 level?

[For the purposes of this question, treat dummy features for categorical variables as separate features.]

```
glm_fit = glm(readmitted ~ ., family = "binomial", data = readmission_train)
summary(glm_fit)
```

```
##
## Call:
## glm(formula = readmitted ~ ., family = "binomial", data = readmission_train)
```

```
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.760  -0.522  -0.444  -0.372   2.605
##
## Coefficients:
##                                                        Estimate Std. Error
## (Intercept)                                           -1.46e+01   6.35e+02
## raceAsian                                             -6.34e-01   7.54e-01
## raceCaucasian                                          1.22e-01   1.22e-01
## raceHispanic                                          -4.03e-01   4.15e-01
## raceOther                                             -1.35e-01   4.22e-01
## genderMale                                             2.04e-02   9.19e-02
## age_group20-59                                         4.10e-01   7.41e-01
## age_group60-79                                         4.85e-01   7.41e-01
## age_group80+                                           4.62e-01   7.47e-01
## num_outpatient                                         3.45e-02   3.22e-02
## num_inpatient                                          2.25e-01   3.00e-02
## num_emergency                                          6.70e-02   4.04e-02
## num_medications                                        4.47e-03   7.19e-03
## num_diagnoses                                          4.87e-02   2.80e-02
## adm_sourceOther                                       -4.41e-01   2.11e-01
## adm_sourcePhysician Referral                          -7.14e-02   1.59e-01
## adm_sourceTransfer from Home Health                   -1.24e-01   2.77e-01
## adm_typeEmergency                                      3.24e-02   1.87e-01
## adm_typeOther                                          2.60e-02   2.43e-01
## adm_typeUrgent                                         1.70e-01   1.64e-01
## time_in_hospital                                      -6.56e-03   1.76e-02
## num_lab_procedures                                     4.14e-04   2.71e-03
## num_procedures                                         6.54e-02   2.95e-02
## dischargeDischarged to home with Home Health Service   6.45e-02   1.44e-01
## dischargeDischarged/Transferred to SNF                 4.36e-01   1.35e-01
## dischargeOther                                         2.59e-01   1.32e-01
## max_glu_serum>300                                      2.67e-01   5.08e-01
## max_glu_serumNone                                     -3.84e-02   4.02e-01
## max_glu_serumNorm                                      1.40e-01   4.32e-01
## A1Cresult>8                                           -4.14e-01   2.97e-01
## A1CresultNone                                         -8.18e-02   2.40e-01
## A1CresultNorm                                         -2.04e-01   3.13e-01
## med_changedYes                                        -1.23e-01   1.70e-01
## med_prescribedYes                                     -4.99e-02   1.69e-01
## insulinNo                                             -4.65e-01   2.31e-01
## insulinSteady                                         -3.21e-01   1.73e-01
## insulinUp                                             -4.02e-01   1.78e-01
## metforminNo                                           -7.28e-01   4.69e-01
## metforminSteady                                       -7.78e-01   4.72e-01
## metforminUp                                            1.30e-01   6.16e-01
## glimepirideNo                                         -8.15e-01   8.39e-01
## glimepirideSteady                                     -6.44e-01   8.57e-01
## glimepirideUp                                         -1.42e+01   3.56e+02
## glipizideNo                                           -6.05e-01   5.77e-01
## glipizideSteady                                       -3.41e-01   5.76e-01
## glipizideUp                                            1.57e-01   7.09e-01
## glyburideNo                                            3.74e-01   7.61e-01
## glyburideSteady                                       4.78e-01   7.62e-01
## glyburideUp                                           5.81e-01   8.69e-01
## pioglitazoneNo                                       -1.43e-01   1.10e+00
## pioglitazoneSteady                                   -6.30e-01   1.12e+00
## pioglitazoneUp                                        8.18e-01   1.30e+00
## rosiglitazoneNo                                       1.37e+01   6.35e+02
## rosiglitazoneSteady                                   1.34e+01   6.35e+02
## rosiglitazoneUp                                       1.36e+01   6.35e+02
##                                                      z value Pr(>|z|)
## (Intercept)                                            -0.02   0.9816
## raceAsian                                              -0.84   0.4004
## raceCaucasian                                           1.00   0.3157
## raceHispanic                                           -0.97   0.3313
## raceOther                                              -0.32   0.7481
## genderMale                                              0.22   0.8245
## age_group20-59                                          0.55   0.5799
## age_group60-79                                          0.65   0.5131
## age_group80+                                            0.62   0.5364
## num_outpatient                                          1.07   0.2830
## num_inpatient                                           7.51  6.1e-14 ***
## num_emergency                                           1.66   0.0977 .
## num_medications                                         0.62   0.5345
## num_diagnoses                                           1.74   0.0814 .
## adm_sourceOther                                        -2.09   0.0363 *
## adm_sourcePhysician Referral                           -0.45   0.6531
## adm_sourceTransfer from Home Health                    -0.45   0.6534
## adm_typeEmergency                                       0.17   0.8628
## adm_typeOther                                           0.11   0.9146
## adm_typeUrgent                                          1.03   0.3026
## time_in_hospital                                       -0.37   0.7094
## num_lab_procedures                                      0.15   0.8784
## num_procedures                                          2.22   0.0265 *
## dischargeDischarged to home with Home Health Service    0.45   0.6551
## dischargeDischarged/Transferred to SNF                  3.24   0.0012 **
## dischargeOther                                          1.97   0.0494 *
## max_glu_serum>300                                       0.53   0.5990
## max_glu_serumNone                                      -0.10   0.9239
## max_glu_serumNorm                                       0.33   0.7452
## A1Cresult>8                                            -1.40   0.1624
## A1CresultNone                                          -0.34   0.7331
## A1CresultNorm                                          -0.65   0.5149
## med_changedYes                                         -0.72   0.4699
## med_prescribedYes                                      -0.30   0.7672
## insulinNo                                              -2.01   0.0441 *
## insulinSteady                                          -1.85   0.0643 .
## insulinUp                                              -2.26   0.0237 *
## metforminNo                                            -1.55   0.1201
## metforminSteady                                        -1.65   0.0997 .
## metforminUp                                             0.21   0.8325
## glimepirideNo                                          -0.97   0.3312
## glimepirideSteady                                      -0.75   0.4519
## glimepirideUp                                          -0.04   0.9683
## glipizideNo                                            -1.05   0.2952
## glipizideSteady                                        -0.59   0.5541
## glipizideUp                                            0.22   0.8249
## glyburideNo                                            0.49   0.6228
## glyburideSteady                                        0.63   0.5305
## glyburideUp                                            0.67   0.5040
## pioglitazoneNo                                        -0.13   0.8960
## pioglitazoneSteady                                    -0.57   0.5719
## pioglitazoneUp                                         0.63   0.5281
## rosiglitazoneNo                                        0.02   0.9828
## rosiglitazoneSteady                                    0.02   0.9831
## rosiglitazoneUp                                        0.02   0.9829
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3617.1  on 4999  degrees of freedom
## Residual deviance: 3448.9  on 4945  degrees of freedom
## AIC: 3559
##
## Number of Fisher Scoring iterations: 14
```

Based on this summary, seven total features are significant at the 0.05 level. Among these, `num_inpatient` and `dischargeDischarged/Transferred to SNF` are also significant at the 0.01 level.

5. Based on the summary above, insulin seems to be the medication most associated with readmission. What is the appropriate statistical test of the null hypothesis that none of the *other* medications (namely, `metformin`, `glimepiride`, `glipizide`, `glyburide`, `pioglitazone`, and `rosiglitazone`) are associated with `readmitted` while controlling for all of the other features? Carry out this test, and state the resulting p-value and the conclusion you would make at the 0.05 significance level.

The appropriate statistical test is the likelihood ratio test (LRT), which we carry out below using `anova`:

```
glm_fit_partial = glm(readmitted ~ .-metformin-glimepiride-glipizide-glyburide-
                        pioglitazone-rosiglitazone,
                      family = "binomial", data = readmission_train)
anova(glm_fit_partial, glm_fit, test = "LRT")
```

```
## Analysis of Deviance Table
##
## Model 1: readmitted ~ (race + gender + age_group + num_outpatient + num_inpatient +
##     num_emergency + num_medications + num_diagnoses + adm_source +
##     adm_type + time_in_hospital + num_lab_procedures + num_procedures +
##     discharge + max_glu_serum + A1Cresult + med_changed + med_prescribed +
##     insulin + metformin + glimepiride + glipizide + glyburide +
##     pioglitazone + rosiglitazone) - metformin - glimepiride -
##     glipizide - glyburide - pioglitazone - rosiglitazone
## Model 2: readmitted ~ race + gender + age_group + num_outpatient + num_inpatient +
##     num_emergency + num_medications + num_diagnoses + adm_source +
##     adm_type + time_in_hospital + num_lab_procedures + num_procedures +
##     discharge + max_glu_serum + A1Cresult + med_changed + med_prescribed +
##     insulin + metformin + glimepiride + glipizide + glyburide +
##     pioglitazone + rosiglitazone
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      4963       3476
## 2      4945       3449 18     26.8    0.083 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The resulting p-value is 0.083. Therefore, at the 0.05 level, we cannot reject the null hypothesis that none of these other drugs are associated with `readmitted`.

6. Suppose patient A and patient B match on all features, except patient A has `insulin` = "Steady" while patient B has `insulin` = "Up". Based on the fitted coefficients from question 4 above, what is the relationship between the predicted odds of `readmitted` for patient B with respect to the corresponding odds for patient A? Express your answer quantitatively.

The coefficients for `insulinSteady` and `insulinUp` are -0.321 and -0.402, respectively. Thus the log-odds of `readmitted` for Patient B are 0.402-0.321 = 0.081 lower than that for Patient A. Therefore, the odds for Patient B are a factor of exp(-0.081) = 0.92 times that of Patient A.

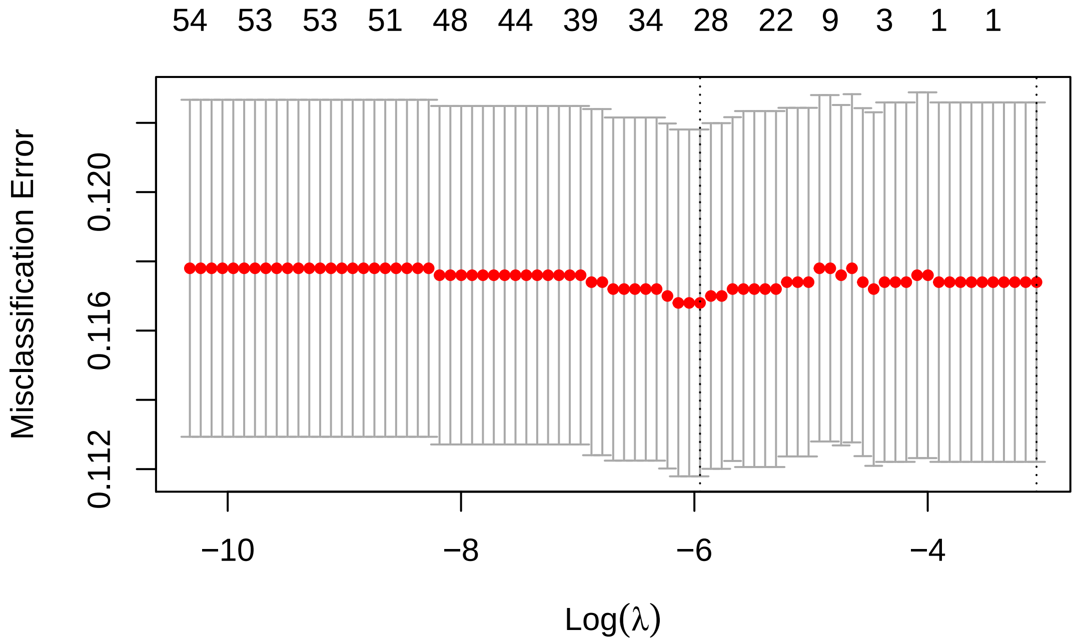
3 Prediction via (lasso) logistic regression

7.

i. Fit a 10-fold cross-validated logistic lasso regression to the training data, using the misclassification error for cross-validation. Produce the cross-validation plot. Why is the left-most number across the top of the CV plot 54, rather than 26, the latter being the total number of features in the data?

```r
# run cross-validated logistic lasso regression
set.seed(3)
X_scaled_train = scale(model.matrix(readmitted ~ ., data = readmission_train)[,-1])
Y_train = readmission_train %>% pull(readmitted)
glmnet_fit = cv.glmnet(x = X_scaled_train, y = Y_train, nfolds = 10,
                       family = "binomial", type.measure = "class",
                       standardize = FALSE, alpha = 1)

# produce CV plot
plot(glmnet_fit)
```



54 is the total number of features after we have converted the categorical variables to binary dummies.

ii. How many nonzero features (not counting the intercept term) are selected when `lambda = lambda.min` and when `lambda = lambda.1se`?

```
# compute number of nonzero features for lambda = lambda.1se and lambda = lambda.min
glmnet_fit$nzero[glmnet_fit$lambda == glmnet_fit$lambda.1se]
```

```
## s0
##  0
```

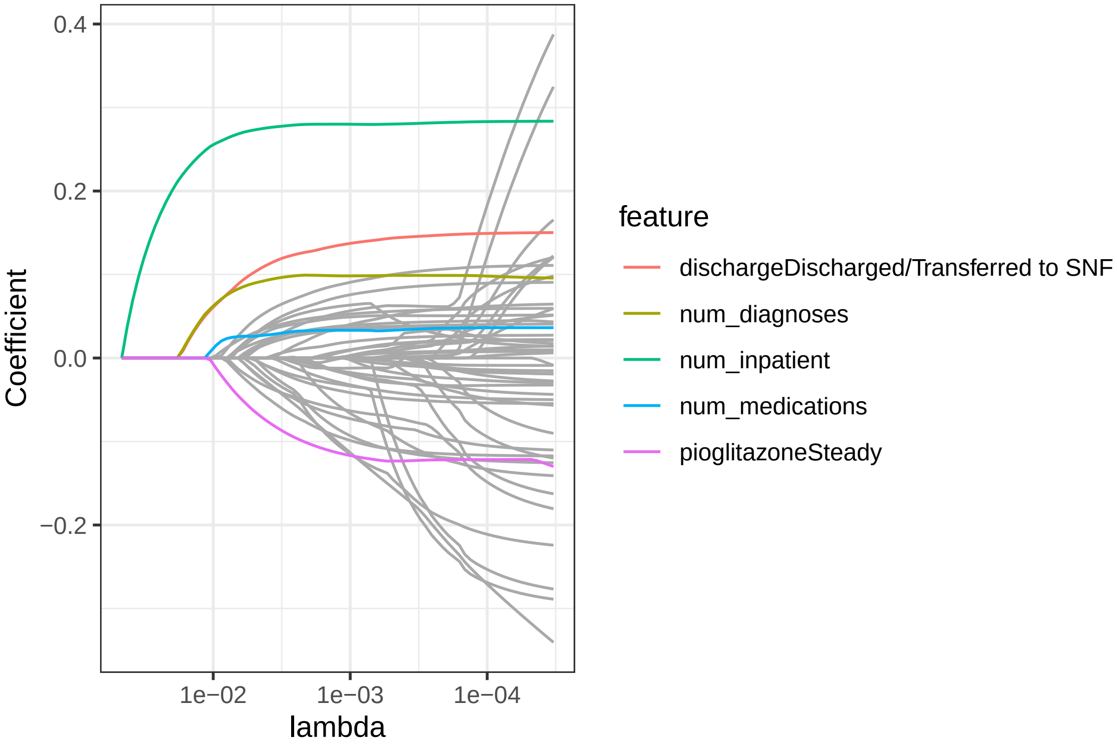
```
glmnet_fit$nzero[glmnet_fit$lambda == glmnet_fit$lambda.min]
```

```
## s31
##  29
```

There are no nonzero variables for `lambda = lambda.1se` and 29 nonzero variables for `lambda = lambda.min`.

iii. Code is provided below to produce the lasso trace plot for `lambda = lambda.min`. Based on this plot, which feature is the first to have a nonzero coefficient?

```
# produce trace plot
plot_glmnet(glmnet_fit, lambda = NA, features_to_plot = 5)
```



Based on the lasso trace plot, we see that `num_inpatient` is the first variable to get a nonzero coefficient.

8. Define vectors `logreg_probabilities` and `glmnet_probabilities` containing the predicted probabilities for each of the test observations under the logistic model and the logistic lasso model (the latter with lambda = lambda.min), respectively.

```
X_scaled_test = scale(model.matrix(readmitted ~ ., data = readmission_test)[,-1])
glmnet_probabilities = predict(glmnet_fit, newx = X_scaled_test,
                               s = "lambda.min", type = "response")[,1]
logreg_probabilities = predict(glm_fit, newdata = readmission_test, type = "response")
```

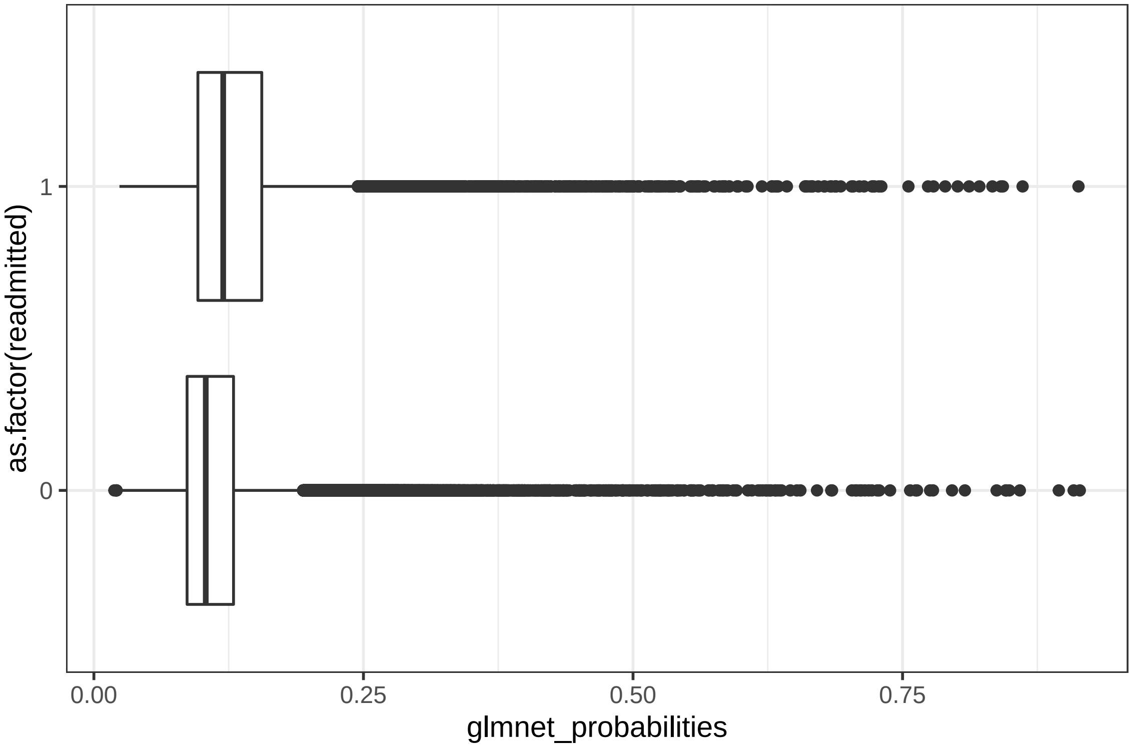
Let's put these together into a data frame, together with the test response:

```
predictions = tibble(glmnet_probabilities,
                     logreg_probabilities,
                     readmitted = readmission_test %>% pull(readmitted))
```

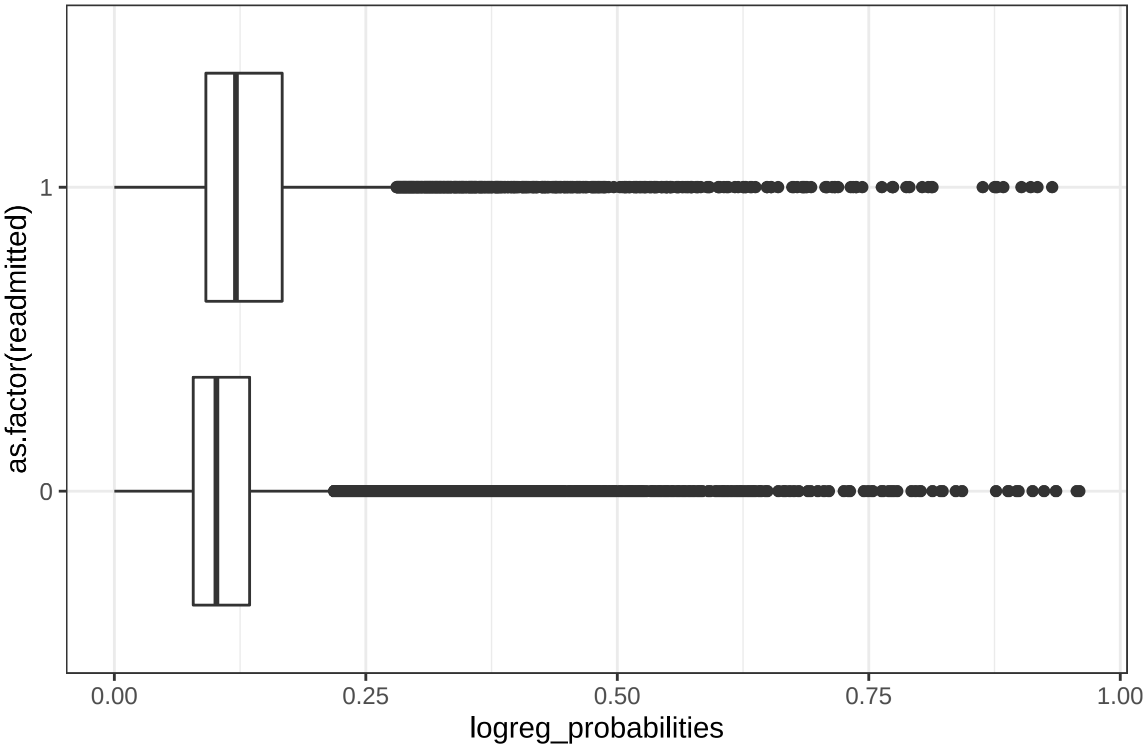
9. For each of the above methods (logistic regression and lasso logistic regression with lambda = lambda.min), make a plot to compare the predicted probabilities to the true binary responses. You should end up with two separate plots. Do the predicted probabilities tend to be larger for test observations for which `readmitted` = 1?

[Hint: use `as.factor(readmitted)`.]

```
predictions %>% ggplot(aes(x = glmnet_probabilities, y = as.factor(readmitted))) +
  geom_boxplot() + theme_bw()
```



```
predictions %>% ggplot(aes(x = logreg_probabilities, y = as.factor(readmitted))) +
  geom_boxplot() + theme_bw()
```



Yes, for both methods the predicted probabilities tend to be larger for test observations for which `readmitted` = 1.

10. Use `mutate` to add two columns to `predictions` called `logreg_predictions` and `glmnet_predictions`, which are the binary classifications of these two methods based on thresholding the probabilities at 0.5 [Hint: `as.numeric` converts logical variables to binary numeric variables]. Then, `summarise` the resulting data frame to obtain the misclassification error of both methods. How do these misclassification errors compare? In what sense is this conclusion consistent with the CV plot from part 8?

```
predictions %>%
  mutate(logreg_predictions = as.numeric(logreg_probabilities > 0.5),
         glmnet_predictions = as.numeric(glmnet_probabilities > 0.5)) %>%
  summarise(logreg_misclass_error = mean(logreg_predictions != readmitted),
            glmnet_misclass_error = mean(glmnet_predictions != readmitted))
```

```
## # A tibble: 1 x 2
##   logreg_misclass_error glmnet_misclass_error
##                   <dbl>                 <dbl>
## 1                 0.112                 0.112
```

We find that both misclassification errors are 0.112, so these methods have the same misclassification error. This conclusion is consistent with the CV plot because the CV curve was relatively flat (with large standard errors), so in this case varying lambda does not have much effect on the misclassification error. In particular, un-penalized logistic regression (whose CV error is at the left-hand endpoint of the CV plot) and penalized logistic regression (whose CV error is at the left-hand dashed line in the CV plot) have similar CV errors.