C3M1_peer_reviewed

June 22, 2023

1 C3M1: Peer Reviewed Assignment

1.0.1 Outline:

The objectives for this assignment:

1. Apply Binomial regression methods to real data.
2. Understand how to analyze and interpret binomial regression models.
3. Flex our math skills by determining whether certain distributions are members of the exponential family.

General tips:

1. Read the questions carefully to understand what is being asked.
2. This work will be reviewed by another human, so make sure that you are clear and concise in what your explanations and answers.

[10]:

```
# Load required libraries
library(tidyverse)
library(dplyr)
```

```
  Attaching packages                                        tidyverse
1.3.0

  ggplot2 3.3.0       purrr   0.3.4
  tibble  3.2.1       dplyr   1.1.2
  tidyr   1.0.2       stringr 1.4.0
  readr   1.3.1       forcats 0.5.0

  Conflicts
tidyverse_conflicts()
  dplyr::filter() masks stats::filter()
  dplyr::lag()    masks stats::lag()
```

1.1 Problem 1: Binomial (Logistic) Regression

The National Institute of Diabetes and Digestive and Kidney Diseases conducted a study of 768 adult female Pima Indians living near Phoenix, AZ. The purpose of the study was to investigate the factors related to diabetes.

Before we analyze these data, we should note that some have raised ethical issues with its collection and popularity in the statistics and data science community. We should think seriously about these concerns. For example, Maya Iskandarani wrote a brief piece on consent and privacy concerns raised by this dataset. After you familarize yourself with the data, we'll then turn to these ethical concerns.

First, we'll use these data to get some practice with GLM and Logistic regression.

[11]:

```
# Load the data
pima = read.csv("pima.txt", sep="\t")
# Here's a description of the data: https://rdrr.io/cran/faraway/man/pima.html
head(pima)
```

A data.frame: 6 × 9

	pregnant <int>	glucose <int>	diastolic <int>	triceps <int>	insulin <int>	bmi <dbl>	diabetes <dbl>	age <int>	test <int>
1	6	148	72	35	0	33.6	0.627	50	1
2	1	85	66	29	0	26.6	0.351	31	0
3	8	183	64	0	0	23.3	0.672	32	1
4	1	89	66	23	94	28.1	0.167	21	0
5	0	137	40	35	168	43.1	2.288	33	1
6	5	116	74	0	0	25.6	0.201	30	0

1.1.1 1. (a) Data Cleaning? What about Data Scrubbing? Data Sterilizing?

This is a real data set, which means that there's likely going to be gaps and missing values in the data. Before doing any modeling, we should inspect the data and clean it if necesary.

Perform simple graphical and numerical summaries of the data. Pay attention for missing or nonsensical values. Can you find any obvious irregularities? If so, take appropriate steps to correct these problems. In the markdown cell, specify what cleaning you did and why you did it.

Finally, split your data into training and test sets. Let the training set contain 80% of the rows and the test set contain the remaining 20%.

[12]:

```
# Your Code Here
head(pima, 10)

#Let's first see if there is any missing values
sum(is.na(pima))
#Great, no missing values in this data set, let's move on to the zeros values␣
 ↪in the columns that does not make sense.
print(paste("glucose: ", sum(pima$glucose == 0)))
print(paste("diastolic: ", sum(pima$diastolic == 0)))
print(paste("triceps: ", sum(pima$triceps == 0)))
print(paste("insulin: ", sum(pima$insulin == 0)))
print(paste("bmi: ", sum(pima$bmi == 0)))
# lets visualize it
```

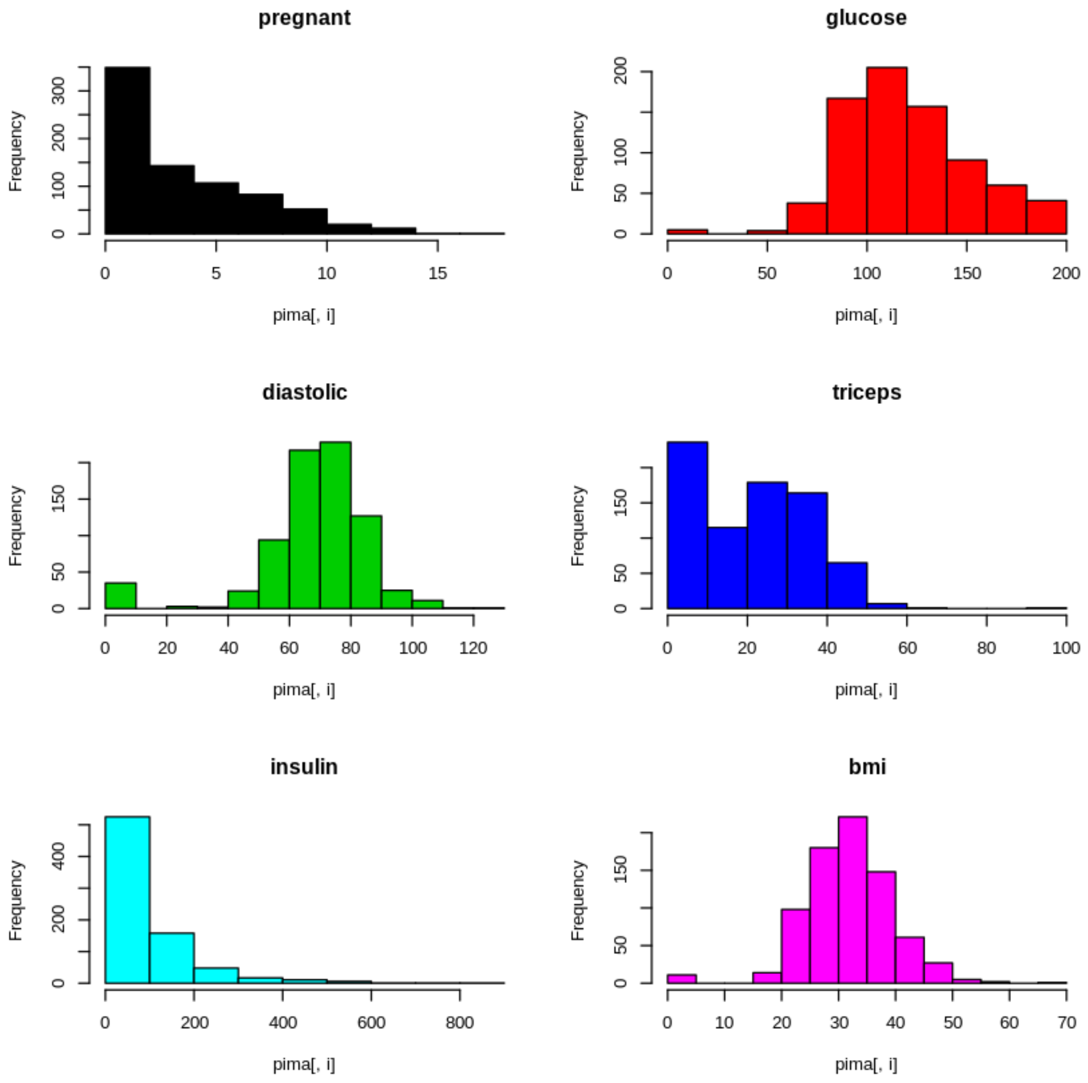
```r
par(mfrow=c(3,2))
for (i in 1:9) hist(pima[,i], col = i, main = names(pima)[i])
```

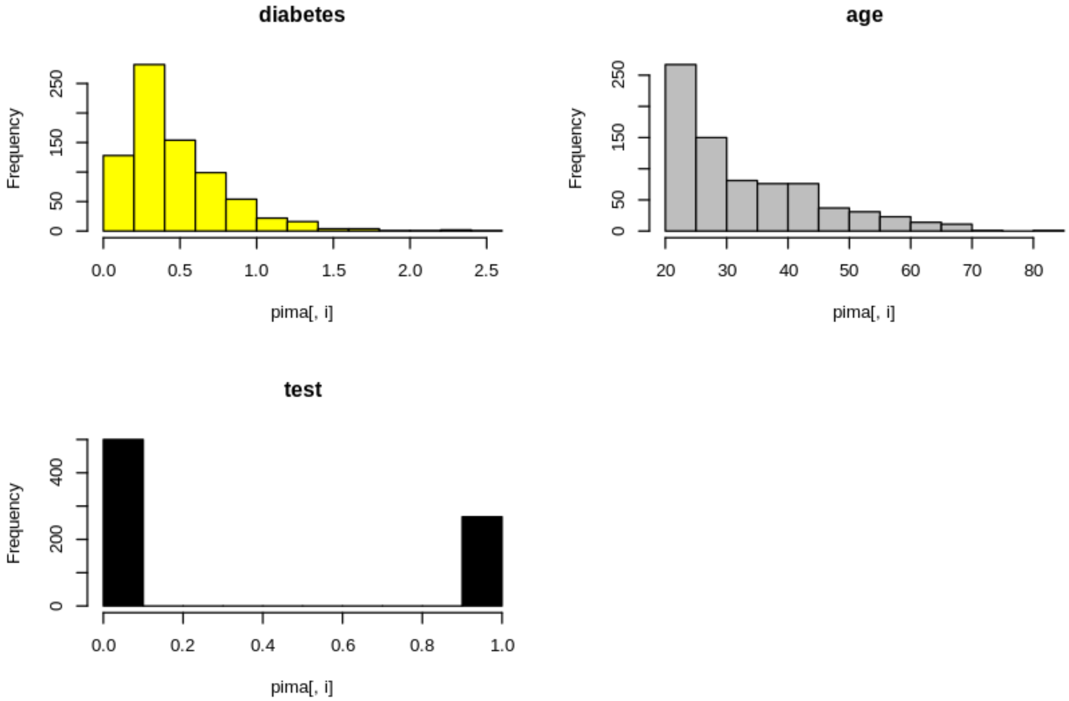
A data.frame: 10 × 9

	pregnant	glucose	diastolic	triceps	insulin	bmi	diabetes	age	test
	<int>	<int>	<int>	<int>	<int>	<dbl>	<dbl>	<int>	<int>
1	6	148	72	35	0	33.6	0.627	50	1
2	1	85	66	29	0	26.6	0.351	31	0
3	8	183	64	0	0	23.3	0.672	32	1
4	1	89	66	23	94	28.1	0.167	21	0
5	0	137	40	35	168	43.1	2.288	33	1
6	5	116	74	0	0	25.6	0.201	30	0
7	3	78	50	32	88	31.0	0.248	26	1
8	10	115	0	0	0	35.3	0.134	29	0
9	2	197	70	45	543	30.5	0.158	53	1
10	8	125	96	0	0	0.0	0.232	54	1

0

```
[1] "glucose:  5"
[1] "diastolic:  35"
[1] "triceps:  227"
[1] "insulin:  374"
[1] "bmi:  11"
```


pregnant
Frequency
0
100
200
300
0
5
10
15
pima[, i]
glucose
Frequency
0
50
100
200
0
50
100
150
200
pima[, i]
diastolic
Frequency
0
50
150
0
20
40
60
80
100
120
pima[, i]
triceps
Frequency
0
50
150
0
20
40
60
80
100
pima[, i]
insulin
Frequency
0
200
400
0
200
400
600
800
pima[, i]
bmi
Frequency
0
50
150
0
10
20
30
40
50
60
70
pima[, i]



[13]:

```r
#let's deal with the zeros numbers in those columns
zeros <- c('glucose', 'diastolic', 'triceps', 'bmi', 'insulin')
pima[zeros][pima[zeros]==0] = NA
pima = na.omit(pima)

# Convert 'test' column to factor
pima <- pima %>%
  mutate(test = as.factor(test))

summary(pima)
nrow(pima)
```

```
    pregnant          glucose        diastolic          triceps
```

```
 Min.   : 0.000   Min.   : 56.0   Min.   : 24.00   Min.   : 7.00  
 1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.:21.00  
 Median : 2.000   Median :119.0   Median : 70.00   Median :29.00  
 Mean   : 3.301   Mean   :122.6   Mean   : 70.66   Mean   :29.15  
 3rd Qu.: 5.000   3rd Qu.:143.0   3rd Qu.: 78.00   3rd Qu.:37.00  
 Max.   :17.000   Max.   :198.0   Max.   :110.00   Max.   :63.00  
    insulin            bmi           diabetes            age        test   
 Min.   : 14.00   Min.   :18.20   Min.   :0.0850   Min.   :21.00   0:262  
 1st Qu.: 76.75   1st Qu.:28.40   1st Qu.:0.2697   1st Qu.:23.00   1:130  
 Median :125.50   Median :33.20   Median :0.4495   Median :27.00          
 Mean   :156.06   Mean   :33.09   Mean   :0.5230   Mean   :30.86          
 3rd Qu.:190.00   3rd Qu.:37.10   3rd Qu.:0.6870   3rd Qu.:36.00          
 Max.   :846.00   Max.   :67.10   Max.   :2.4200   Max.   :81.00          
```

392

[14]:

```
#Let's split the data
set.seed(1994)

n = floor(0.8 * nrow(pima))
index = sample(seq_len(nrow(pima)), size = n)

train = pima[index, ]
test = pima[-index, ]
```

1.1.2 1. (b) Initial GLM modelling

Our data is clean and we're ready to fit! What kind of model should we use to fit these data? Notice that the `test` variable is either 0 or 1, for whether the individual tested positive for diabetes. Because `test` is binary, we should use logistic regression (which is a kind of binomial regression).

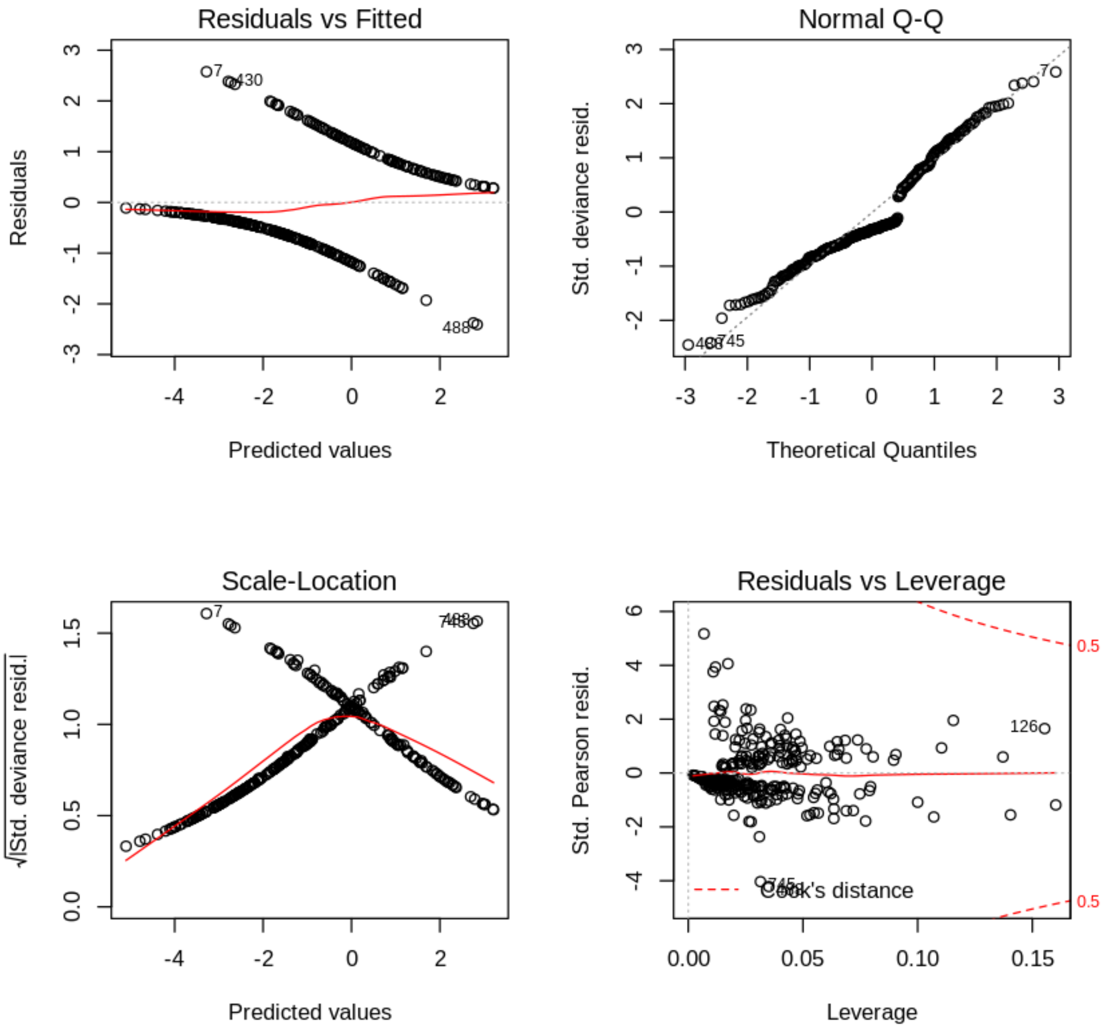
Fit a model with `test` as the response and all the other variables as predictors. Can you tell whether this model fits the data?

[15]:

```
# Your Code Here
mod_pima = glm(test ~ ., data = train, family = binomial)
summary(mod_pima)
par(mfrow = c(2,2))
plot(mod_pima)
```

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

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-2.4081  -0.6510  -0.3212   0.6222   2.5756  
```

```
Coefficients:
              Estimate Std. Error z value Pr(>|z|)    
(Intercept) -10.265419   1.358116  -7.559 4.08e-14 ***
pregnant      0.069203   0.064156   1.079  0.28074    
glucose       0.041050   0.006602   6.218 5.04e-10 ***
diastolic    -0.001437   0.013108  -0.110  0.91273    
triceps       0.041582   0.019349   2.149  0.03163 *  
insulin      -0.001325   0.001540  -0.860  0.38955    
bmi           0.055985   0.030271   1.849  0.06439 .  
diabetes      1.615317   0.516936   3.125  0.00178 ** 
age           0.011443   0.020031   0.571  0.56783    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 400.73  on 312  degrees of freedom
Residual deviance: 270.73  on 304  degrees of freedom
AIC: 288.73

Number of Fisher Scoring iterations: 5
```



When dealing with binary responses ($Y = 0, 1$) instead of a broader range of values, the residuals will not conform to a normal distribution and the deviance will not follow a chi-squared distribution. As a result, traditional tests for model fit cannot be applied. In such cases, a common approach is to split the data into a training set and a test set. The model can then be evaluated by examining its performance in predicting values on the test set. This provides a practical measure of how well the model performs in practice.

1.1.3 1. (c) Remember Bayes

A quick analytical interlude.

Is diastolic blood pressure significant in the regression model? Do women who test positive have higher diastolic blood pressures? Explain the distinction between the two questions and discuss why the answers are only apparently contradictory.

[16]:

```
# Your Code Here
summary(mod_pima)
lm_diastolic = lm(diastolic ~ test, data = train)
summary(lm_diastolic)
```

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

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4081  -0.6510  -0.3212   0.6222   2.5756

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -10.265419   1.358116  -7.559 4.08e-14 ***
pregnant      0.069203   0.064156   1.079  0.28074
glucose       0.041050   0.006602   6.218 5.04e-10 ***
diastolic    -0.001437   0.013108  -0.110  0.91273
triceps       0.041582   0.019349   2.149  0.03163 *
insulin      -0.001325   0.001540  -0.860  0.38955
bmi           0.055985   0.030271   1.849  0.06439 .
diabetes      1.615317   0.516936   3.125  0.00178 **
age           0.011443   0.020031   0.571  0.56783
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 400.73  on 312  degrees of freedom
Residual deviance: 270.73  on 304  degrees of freedom
AIC: 288.73

Number of Fisher Scoring iterations: 5



Call:
lm(formula = diastolic ~ test, data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-44.889  -8.889   1.111   9.111  37.111

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  68.8889     0.8825   78.06  < 2e-16 ***
```

```
test1          5.1111     1.5165    3.37 0.000845 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 12.7 on 311 degrees of freedom
Multiple R-squared:  0.03524,Adjusted R-squared:  0.03214
F-statistic: 11.36 on 1 and 311 DF,  p-value: 0.0008452
```

From the lm model we can see that women who test positive do have higher diastolic blood pressures, but in the logistic regression model the diastolic blood pressure is not significant. There are two distinct questions that involve conditional probabilities: one relates to the outcome of the test given a certain diastolic pressure, while the other pertains to the diastolic blood pressure given a specific test result. It is important to note that these two questions are not equivalent, as they involve different conditional probabilities. According to Bayes’ theorem, the relationship between these conditional probabilities is not symmetrical, and thus they cannot be treated interchangeably.

1.1.4 1. (d) GLM Interpretation

We’ve seen so many regression summaries up to this point, how is this one different from all the others? Well, to really understand any model, it can be helpful to loop back and plug the fitted results back into the model’s mathematical form.

Explicity write out the equation for the binomial regression model that you fit in (b). Then, in words, explain how a 1 unit change of `glucose` affects `test`, assuming all other predictors are held constant.

[21]:
```
# Your Code Here
summary(mod_pima)
```

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

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.8166  -0.6627  -0.3728   0.6588   2.5346

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.010e+01  1.373e+00  -7.358 1.87e-13 ***
pregnant     1.382e-01  6.213e-02   2.225  0.02609 *
glucose      3.482e-02  6.211e-03   5.607 2.06e-08 ***
diastolic   -3.315e-04  1.350e-02  -0.025  0.98041
triceps      1.142e-02  1.899e-02   0.602  0.54736
insulin     -3.482e-04  1.472e-03  -0.237  0.81304
bmi          8.029e-02  3.044e-02   2.638  0.00835 **
diabetes     1.152e+00  4.789e-01   2.406  0.01611 *
```

```
age          2.710e-02  2.027e-02   1.337  0.18120
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 398.80  on 313  degrees of freedom
Residual deviance: 277.41  on 305  degrees of freedom
  (300 observations deleted due to missingness)
AIC: 295.41

Number of Fisher Scoring iterations: 5
```

$\eta = \log(\hat{p}_1 - \hat{p}) = \beta_0 + \beta_1 \cdot \text{pregnant} + \beta_2 \cdot \text{glucose} + \beta_3 \cdot \text{diastolic} + \beta_4 \cdot \text{triceps} + \beta_5 \cdot \text{insulin} + \beta_6 \cdot \text{bmi} + \beta_7 \cdot \text{diabetes} + \beta_8 \cdot \text{age}$ When considering other predictors, an increase of one unit in glucose levels corresponds to a 0.03 increase in the log-odds of a positive test. Alternatively, when adjusting for other predictors, a one-unit increase in glucose levels results in an approximately 1.03 increase in the odds of success.

1.1.5 1. (e) GLM Prediction

One of the downsides of Logistic Regression is that there isn't an easy way of evaulating the goodness of fit of the model without predicting on new data. But, if we have more data to test with, then there are many methods of evaluation to use. One of the best tools are confusion matrices, which (despite the name) are actually not that hard to understand.

A confusion matrix compares the predicted outcomes of a Logistic Regression Model (or any classification model) with the actual classifications. For binary classification, it is a 2×2 matrix where the rows are the models' predicted outcome and the columns are the actual classifications. An example is displayed below.

	True	False
1	103	37
0	55	64

In the example, we know the following information: * The [1,1] cell is the number of datapoints that were correctly predicted to be 1. The value (103) is the number of True Positives (TP). * The [2,2] cell is the number of datapoints that were correctly predicted to be 0. The value is the number of True Negatives (TN). * The [1, 2] cell is the number of datapoints that were predicted to be 1 but where actually 0. This is the number of False Positives (FP), also called Type I error. In the context of our diabetes dataset, this would mean our model predicted that the person would have diabetes, but they actually did not. * The [2, 1] cell is the number of datapoints that were predicted to be 0 but where actually 1. This is the number of False Negatives (FN), also called Type 2 error. In the context of our diabetes dataset, this would mean our model predicted that the person would not have diabetes, but they actually did have diabetes.

Use your model to predict the outcomes of the test set. Then construct a confusion matrix for these predictions and display the results.

[31]:

```
# Your Code Here
#Let's predict the binary outcome based on the logistic regression model's␣
 ↪predicted probabilities. If a predicted probability
#is greater than 0.5, the corresponding element in the output vector prob will␣
 ↪be 1,
#indicating a positive outcome. Otherwise, it will be 0, indicating a negative␣
 ↪outcome.

prob = ifelse(predict.glm(mod_pima, type = "response", test, na.rm = TRUE) > 0.
 ↪5, 1, 0)
tp = sum(prob == 1 & as.numeric(levels(test$test))[test$test] == 1);
fp= sum(prob == 1 & as.numeric(levels(test$test))[test$test] == 0);
fn= sum(prob == 0 & as.numeric(levels(test$test))[test$test] == 1);
tn = sum(prob == 0 & as.numeric(levels(test$test))[test$test] == 0);


# Create the confusion matrix
confusion_matrix <- table( Predicted = as.factor(prob), Actual = as.
 ↪factor(test$test))

# Print the confusion matrix
print(confusion_matrix)
```

```
         Actual
Predicted  0  1
        0 49 15
        1  6  9
```

1.1.6 1. (f) Evaluation Statistics

Using the four values from the confusion matrix, we can construct evaulation statistics to get a numerical approximation for our model's performance. Spend some time researching accuracy, precision, recall and F score.

Calculate these values for your model's predictions on the test set. Clearly display your results. How well do you think your model fits the data?

[41]:

```
# Your Code Here
accuracy = (tp+tn)/(tp+tn+fp+fn);
precision = tp/(tp+fp);
recall = tp/(tp+fn);

accuracy
precision
```

```
recall

F = (2*precision*recall)/(precision + recall)
F
```

0.734177215189873

0.6

0.375

0.461538461538462

The F score, ranging from 0 to 1, offers a comprehensive measure that combines precision and recall. With an F score of 0.46, the performance can be considered poor, because value below 0.5 suggests that the model's ability to balance both aspects is not optimal. However, whether an F score below 0.5 is considered "bad" depends on the specific context and the acceptable level of performance for the given problem. In some cases, a lower F score may still be acceptable depending on the trade-offs and requirements of the application. It is important to consider the specific domain and context when interpreting the significance of an F score below 0.5.

1.1.7 1. (g) Understanding Evaluation Statistics

Answer the following questions in the markdown cell below.

1. Give an example scenario for when accuracy would be a misleading evaulation statistic.
2. Confusion matrices can also be used for non-binary classification problems. Describe what a confusion matrix would look like for a response with 3 levels.
3. You'll have to take our word on the fact (or spend some time researching) that Type I error and Type II error are inversely related. That is, if a model is very good at detecting false positives, then it will be bad at detecting false negatives. In the case of our diabetes dataset, would you prefer a model that overestimates the Type 1 error or overestimates the Type II error. Justify your answer.

1. An example scenario where accuracy would be a misleading evaluation statistic is when dealing with imbalanced datasets. Suppose you have a binary classification problem where the positive class is rare, representing only 5% of the total observations. If you have a classifier that always predicts the negative class, it would achieve an accuracy of 95% simply by predicting the majority class. However, this high accuracy does not indicate good performance, as the classifier fails to correctly identify the positive class, which may be more important or critical in the given context. In such imbalanced scenarios, other evaluation metrics like precision, recall, or F1 score are more informative and reliable.

2. A confusion matrix for a response with three levels would have a square matrix with dimensions corresponding to the number of levels. Let's consider an example with three levels: "Low," "Medium," and "High."

3. In the case of the diabetes dataset, it would be preferable to overestimate the Type II error (false negatives) rather than overestimating the Type I error (false positives).Type I error refers to falsely identifying a person as having diabetes when they do not, while Type II error refers to failing to identify a person with diabetes. In this context, overestimating the Type

II error means that the model might miss identifying some individuals with diabetes, leading to false negatives. The justification for preferring overestimating the Type II error is based on the potential consequences of misclassification. In a medical scenario like diabetes, it is generally more critical to identify individuals with the condition to ensure timely treatment and prevent complications. Missing a diagnosis (Type II error) could lead to delayed or no treatment, negatively impacting the patient's health. On the other hand, overestimating the Type I error (false positives) might result in unnecessary follow-ups or treatments, but it is less detrimental compared to missing a true positive.

Hence, prioritizing a model that overestimates the Type II error would be more beneficial in this case to minimize the risk of missing individuals who actually have diabetes.

1.1.8 1. (h) Ethical Issues in Data Collection

Read Maya Iskandarani's piece on consent and privacy concerns raised by this dataset. Summarize those concerns here.

The specific concerns can be summarized as follows:

1.Complexity of Medical Consent: Radin highlights that medical consent is not a simple matter and goes beyond a one-time agreement. Consent for medical research involving data collection and analysis can have long-term implications that traverse generations. The challenge lies in informing study participants about the potential uses and consequences of their medical data far into the future.

2.Privacy and Accessibility of Data: Radin discusses the case of the Pima Native American tribe and the publicly accessible Pima Indian Diabetes Data set (PIDD). While the data set has been valuable for refining machine learning algorithms to predict and prevent diabetes, it raises privacy concerns. Personal health information, such as blood pressure, BMI, and pregnancy history, is publicly available, raising ethical questions about the accessibility and protection of such sensitive data.

3.Ethical Controversy: The availability of the PIDD and other similar data sets in repositories like the UCI Machine Learning Repository raises ethical controversies. The use of long-term, publicly accessible data creates tensions between the potential benefits of research and the privacy rights and consent of individuals whose data is included in the dataset.

4.Interdisciplinary Considerations: Radin's research brings together multiple disciplines, including medical history, anthropology, bioethics, and data analytics. The interdisciplinary nature of the research allows for a comprehensive exploration of the complex ethical issues at the intersection of medical research, data privacy, and consent.

1.2 Problem 2: Practicing those Math skills

One of the conditions of GLMs is that the "random component" of the data needs to come from the Exponential Family of Distributions. But how do we know if a distribution is in the Exponential Family? Well, we could look it up. Or we could be proper mathematicians and check the answer ourselves! Let's flex those math muscles.

1.2.1 2. (a) But it's in the name...

Show that $Y \sim exponential(\lambda)$, where λ is known, is a member of the exponential family.

If the distribution of a random variable Y, represented as either a probability density function (pdf) or a probability mass function (pmf), can be expressed in the following form:

$$f(y; \theta, \phi) = \exp(y\theta - b(\theta)/a(\phi) + c(y, \phi))$$

If the probability density function (pdf) of random variable Y follows the specific form, then Y is considered to have an exponential distribution:

$$f(y; \lambda) = \lambda e^{-\lambda y} = \exp(\log(\lambda e^{-\lambda y})) = \exp(\log(\lambda) - \lambda y) = \exp(\lambda y - \log(\lambda)/-1 + 0)$$

1.2.2 2. (b) Why can't plants do math? Because it gives them square roots!

Let $Y_i \sim exponential(\lambda)$ where $i \in \{1, \dots, n\}$. Then $Z = \sum_{i=1}^{n} Y_i \sim Gamma(n, \lambda)$. Show that Z is also a member of the exponential family.

$$f(y; n, \lambda) = (\lambda^n/\Gamma(n))y^{n-1}e^{-\lambda y}$$

We can see that the form of the exponential family of distributions is presented