C3M1: Peer Reviewed Assignment

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.

In [2]:
```
# Load required libraries
library(tidyverse)
library(dplyr)
```

```
── Attaching packages ─────────────────────────────────────── tidyver
se 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_con
flicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
```

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 (https://researchblog.duke.edu/2016/10/24/diabetes-and-privacy-meet-big-data/)](https://researchblog.duke.edu/2016/10/24/diabetes-and-privacy-meet-big-data/) 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.

In [3]:
```
# Load the data
pima = read.csv("pima.txt", sep="\t")
# Here's a description of the data: https://rdrr.io/cran/faraway/man/p
head(pima)
```

A data.frame: 6 × 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

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%.

In [4]:
```
# Summary Information
summary(pima)
head(pima)

# Scrutinize
cat("\n Any NANs?", sum(is.na(pima))>0 , "\n")
str(pima)

# Visualize
par(mfrow=c(3,3))
for (i in 1 : dim(pima)[2]){
    categ = names(pima)[i]
    hist(pima[,i], main = categ)
    cat("\n " ,categ ," # of Zeros:",nrow(pima[pima[i]==0]))
    }

# Bad zeros to Means (impute instead of drop)
pima$glucose[pima$glucose == 0] = mean(pima$glucose)
pima$diastolic [pima$diastolic  == 0] = mean(pima$diastolic )
pima$triceps [pima$triceps  == 0] = mean(pima$triceps)
pima$insulin [pima$insulin  == 0] = mean(pima$insulin )
pima$bmi[pima$bmi== 0] = mean(pima$bmi)
pima$age[pima$age== 0] = mean(pima$age)

# Double Check
for (i in 1 : dim(pima)[2]){
    categ = names(pima[i])
    hist(pima[,i], main = categ)
    cat("\n ",categ, " # of Zeros:",nrow(pima[pima$categ==0]))
    }

# Reset environment
par(mfrow=c(1,1))

set.seed(77)
# Train test split
```

```
n = floor(nrow(pima)*.8)
indexed = sample(seq_len(nrow(pima)),n)
train = pima[indexed,]
test = pima[-indexed,]
dim(train)
dim(test)
```

```
    pregnant         glucose        diastolic         triceps     
 Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00  
 1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.: 0.00  
 Median : 3.000   Median :117.0   Median : 72.00   Median :23.00  
 Mean   : 3.845   Mean   :120.9   Mean   : 69.11   Mean   :20.54  
 3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.00   3rd Qu.:32.00  
 Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00  
    insulin           bmi          diabetes           age       
 Min.   :  0.0   Min.   : 0.00   Min.   :0.0780   Min.   :21.00  
 1st Qu.:  0.0   1st Qu.:27.30   1st Qu.:0.2437   1st Qu.:24.00  
 Median : 30.5   Median :32.00   Median :0.3725   Median :29.00  
 Mean   : 79.8   Mean   :31.99   Mean   :0.4719   Mean   :33.24  
 3rd Qu.:127.2   3rd Qu.:36.60   3rd Qu.:0.6262   3rd Qu.:41.00  
 Max.   :846.0   Max.   :67.10   Max.   :2.4200   Max.   :81.00  
      test       
 Min.   :0.000  
 1st Qu.:0.000  
 Median :0.000  
 Mean   :0.349  
 3rd Qu.:1.000  
```

How I Cleaned

Zeros in glucose, diastolic blood pressure, bmi, insulin, tricep skin thickness, or age are discussing a deceased or an imaginary type of patient. We will impute those to their means to study as much data as possible. The alternative, dropping the 0's isn't as good an option because a load of data would be lost.

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?

In [5]:
```
# Build GLM model
dbt = glm(test ~ pregnant + glucose + diastolic + triceps + insulin +
            data = train, family = binomial)
summary(dbt)

# Get p-value for the test
cat("\n P-Value of Model:", 1-pchisq(568.43,605))
cat("\n Double Check P-Value:", pchisq(deviance(dbt),df.residual(dbt),
```

```
Call:
glm(formula = test ~ pregnant + glucose + diastolic + triceps +
    insulin + bmi + diabetes + age, family = binomial, data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6017  -0.7275  -0.3956   0.7402   2.3447

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -9.0234949  0.9045140  -9.976  < 2e-16 ***
pregnant     0.1306249  0.0361952   3.609 0.000307 ***
glucose      0.0388272  0.0044606   8.705  < 2e-16 ***
diastolic   -0.0156356  0.0095374  -1.639 0.101130
triceps     -0.0009725  0.0129289  -0.075 0.940043
insulin     -0.0018674  0.0011205  -1.667 0.095588 .
bmi          0.1101007  0.0204619   5.381 7.42e-08 ***
diabetes     0.7680117  0.3346124   2.295 0.021720 *
age          0.0113059  0.0108880   1.038 0.299090
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 788.83  on 613  degrees of freedom
Residual deviance: 568.43  on 605  degrees of freedom
AIC: 586.43

Number of Fisher Scoring iterations: 5


 P-Value of Model: 0.854061
 Double Check P-Value: 0.8540357
```

This model fits the data for two reasons.

- the AIC is less than the null deviance.
- the P-Value, at levels over .85 alpha, is so larget that we do not have evidence to dispute the null hypothesis that the model fits the data well...

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.

In [8]:
```
# Your Code Here
cat("Model without Diastolic for Comparison (db2): \n")
db2 = glm(test~pregnant + glucose  + triceps + insulin + bmi + diabete
summary(db2)

cat("\n Double Check P-Value:", pchisq(deviance(db2),df.residual(db2),
cat("\n P-Value of model db2:", 1-pchisq(571.14,606), "\n ------------
cat("\n Comparison of Female Diastolic: (only able to tell sex when pr
cat("\n Mean of Diastolic Readings in Pregnant Women:", mean(train$dia
cat("\n Mean of Diastolic Readings: ", mean(train$diastolic) )
cat("\n Mean of Diastolic Readings in Diabetic Pregnant Women:", mean(
cat("\n Mean of Diastolic Readings in non pregnant Diabetics: ", mean(
```

```
Model without Diastolic for Comparison (db2):


Call:
glm(formula = test ~ pregnant + glucose + triceps + insulin +
    bmi + diabetes + age, family = binomial, data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6463  -0.7057  -0.3927   0.7270   2.4437

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -9.587227   0.845953 -11.333  < 2e-16 ***
pregnant     0.125681   0.035877   3.503  0.00046 ***
glucose      0.037678   0.004355   8.651  < 2e-16 ***
triceps     -0.001057   0.012865  -0.082  0.93451
insulin     -0.001607   0.001105  -1.455  0.14565
bmi          0.101085   0.019609   5.155 2.54e-07 ***
diabetes     0.754731   0.333934   2.260  0.02381 *
age          0.007377   0.010583   0.697  0.48575
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 788.83  on 613  degrees of freedom
Residual deviance: 571.14  on 606  degrees of freedom
AIC: 587.14
```

```
Number of Fisher Scoring iterations: 5


 Double Check P-Value: 0.8418425

 P-Value of model db2: 0.8418172
 ------------------------------------------------

 Comparison of Female Diastolic: (only able to tell sex when pregnan
t)
 Mean of Diastolic Readings in Pregnant Women: 72.08291
 Mean of Diastolic Readings:  72.24079
 Mean of Diastolic Readings in Diabetic Pregnant Women: 74.88986
 Mean of Diastolic Readings in non pregnant Diabetics:  74.97859
```

NO:

If significance is measured with an alpha of 5%, diastolic is an inconsequential variable in the model. The P-Value of diastolic in the original model ranges depending on the randomness of the train/test split, .06296 to .1517, the variable is sometimes significant at an alpha of 10%. I'm showing a split with 10.1% pvalue, making it insignificant. To further flesh out significance, we remove diastolic in a second model, db2. In the db2 model the AIC falls marginally and there is only a mild reduction in PValue (from above .85 to .84). Therefore, leaving diastolic in the model is insignificant.

YES:

This problem does seem to conform to bayesian theory $P(A|B) = \frac{P(B|A)*P(A)}{P(B)}$. Women who are diabetic seem to have elevated diastolic pressures when compared to non-diabetic pregnant women. The probability of the outcome seems to be dependent on another variable (in this case pregnancy). However, pregnant women who test positive for diabetes do not have a higher mean diastolic blood pressure than those diabetics who are not pregnant, putting the conclusion in doubt. Finally, this test does not account for the women who are not pregnant (the Pregnant variable does not capture all women in the study). Without a sex variable, we are unable to conclude decisively that there is a bayesian statistical anomoly in female diastolic blood pressures based on pregnancy. The reason comes down to mutual exclusivity, because we are deriving our hypothesis from a variable that doesn't capture all females.

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.

In [22]:
```
# Your Code Here
dbt$coefficients

exp(.38)
```

(Intercept)
-9.04626882367784
pregnant
0.130244332775521
glucose
0.0378040468857092
diastolic
-0.0145270692645603
triceps
-0.00390955574114133
insulin
-0.00167540084381559
bmi
0.111517936586737
diabetes
0.608682378333225
age
0.0156108537478003

1.46228458943422

log-odds of test = -9.046 + pregnant(0.13) + glucose(0.038) + diastolic(-0.015) + triceps(-0.004) + insulin(-0.002) + bmi(0.112) + diabetes(0.609)

odds of test = $e^{(logodds)} = e^{(-9.046+pregnant(0.13)+glucose(0.038)+diastolic(-0.015)+triceps(-0.004)+insulin(-0.002)+bmi(0.1}$

So to add a unit of glucose to the odds is multiplying all other items held constant by $e^{0.38}$ or 1.46, increasing the odds by 46%.

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.

In [84]:
```
# Your Code Here
test$pred = exp(predict(dbt,newdata = test[1:8],type = "link"))
test$preds = 0
# arbitrary cutoff point in predictions...
test$preds[test$pred > .5]=1
#head(test)
tp = nrow(test[test$test == 1 & test$preds == 1, c('bmi','test','preds
fp = nrow(test[test$test == 0 & test$preds == 1, c('bmi','test','preds
tn = nrow(test[test$test == 0 & test$preds == 0, c('bmi','test','preds
fn = nrow(test[test$test == 1 & test$preds == 0, c('bmi','test','preds

conf =  data.frame(x1 = c(0,0),x2 = c(0,0))
rownames(conf) = c('(1)','(0)')
colnames(conf) = c('Positive','Negative')
#conf
conf[1,1] = tp
conf[2,1] = fp
conf[2,2] = tn
conf[1,2] = fn
conf
```

A data.frame: 2 × 2

	Positive	Negative
	<dbl>	<dbl>
(1)	40	16
(0)	27	71

As discussed above, I set the arbitrary cutoff point for classification between [0,1] at .5. This yields more false positives, but reduces the number of false negatives (type I error).

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?

In [85]:
```
# Your Code Here
cat("accuracy : " , tp/(tp+fn) , "% \n")
prec = tp/(tp+fp)
cat("precision : " , prec, "% \n")
recc = (tp+tn)/(tp+fp+tn+fn)
cat("recall : " , recc, "% \n")
cat("f1 score: " , 2*prec*recc/(prec+recc) , "% \n")
```

```
accuracy :  0.7142857 %
precision :  0.5970149 %
recall :  0.7207792 %
f1 score:  0.6530852 %
```

This model is far better than guessing, but leaves a lot to be desired. The precision is below guessing, the model could spit out all zeros and get 65% based on the class structure. To deploy a model with this level of recall and accuracy would not be smart either. However, this model may be useful to trained experts looking for candidates that are probably diabetic more efficiently.

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. When running GLM on datasets with unbalanced classes, accuracy can yield misleading accounts of model efficacy. For instance, 1000 observations with 30 positive cases. If the model returned all zeros (negative), the accuracy would be 97%. This would tell the unwary practitioner that the model is sufficient for deployment when in fact the model is unfunctional. Precision, recall, and ROC-AUC would be better metrics for explaining the efficacy of such a model. With our data above, the positive rate is only 35%. Therefore we should be careful trusting any accuracy score around 65% as it could be a similar (all zeros) model output.
2. A confusion matrix with 3 levels would be a 3x3 grid, with correct predictions counted down the diagonal. The six boxes off the diagonal would be inversely mirrored. Counting up the hits and misses would need to be done from the diagonal upwards or from the diagonal downwards to avoid double counting.
3. As built, I would prefer to have fewer false negatives (person has diabetes and we miss it) because of the health implications. Detecting all of the diabetes cases in the sample would be of the highest importance because medical intervention could extend the life of the patients. Also, a secondary screening of each positive result could be used to filter out the false positives after the fact. Further, there exists no precise way to find false negatives. To find false negative we'd need to double test the whole negative population, wasting resources on mostly healthy patients.

1. (h) Ethical Issues in Data Collection

Read Maya Iskandarani's piece (https://researchblog.duke.edu/2016/10/24/diabetes-and-privacy-meet-big-data/) on consent and privacy concerns raised by this dataset. Summarize those concerns here.

Informed consent is a very complex issue in machine learning and statistical analysis. When gathering personal medical data the issue becomes more tangible. The issue comes to the fore when using the PIDD dataset as we are here. The data was collected from 1964 to 2004 by the NIH, a period 4x longer than the initial consent request for 10 years of study. Worse, the data continues to live on public servers without restricted access in myriad locations online. Although the data benefits society by serving as a benchmark for ML/AI algorithms and exercises, the privacy of the individuals studied is not guarded or restricted. Professor Iskandarani brings attention to the fact that consent in perpetuity empowers the data holders to use the information in any use case, despite many applications being contrary to the nature of the initial consent. However, because initial consent was granted, the PIMA indians have little to no recourse to affect change. Why does this matter? Because every consent to terms, every iphone and android agreement, is similar. Statisticians and ML practitioners can understand these nuances and should be aware of the privacy reality. Awareness is the first step toward a fair and honest use of data in the future.

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.

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.

Exponential Family Definition: $f_y(y|\theta, \phi) = e^{(\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi))}$

Exponential Definition: $f(y; \lambda) = \lambda e^{-\lambda y}$

Mutate to Exponential Family form:

= $e^{log(\lambda)} * e^{-\lambda y}$

= $e^{log(\lambda) - \lambda y}$

= $e^{\frac{\lambda y - log(\lambda)}{-1}}$ <-- required form :)

$\theta = \lambda, \phi = -1$

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

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

Exponential Definition: $f(y;\lambda) = \lambda e^{-\lambda y}$

Gamma Definition: $f(y) = \frac{1}{\Gamma(\alpha)} \beta^{\alpha} y^{\alpha-1} e^{-\beta y}$

Mutate to Exponential Form:

$= e^{log(\frac{1}{\Gamma(\alpha)} \beta^{\alpha} y^{\alpha-1})} e^{-\beta y}$

$= e^{-\beta y + log(\frac{\beta^{\alpha} y^{\alpha-1}}{\Gamma(\alpha)})}$

$= e^{-\beta y + (\alpha-1)log(y) + \alpha log(\beta) - \alpha log(\Gamma)}$

$= e^{(\alpha)log(y) - \beta y + (-log(y)) + \alpha log(\beta) - \alpha log(\Gamma)}$ <-- required form :)

$\theta = (\alpha)log(y)$

In [ ]: