# 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 [1]:
```
# 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()
```

# 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/) 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 [2]:
```
# 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 | 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 $20\%$.

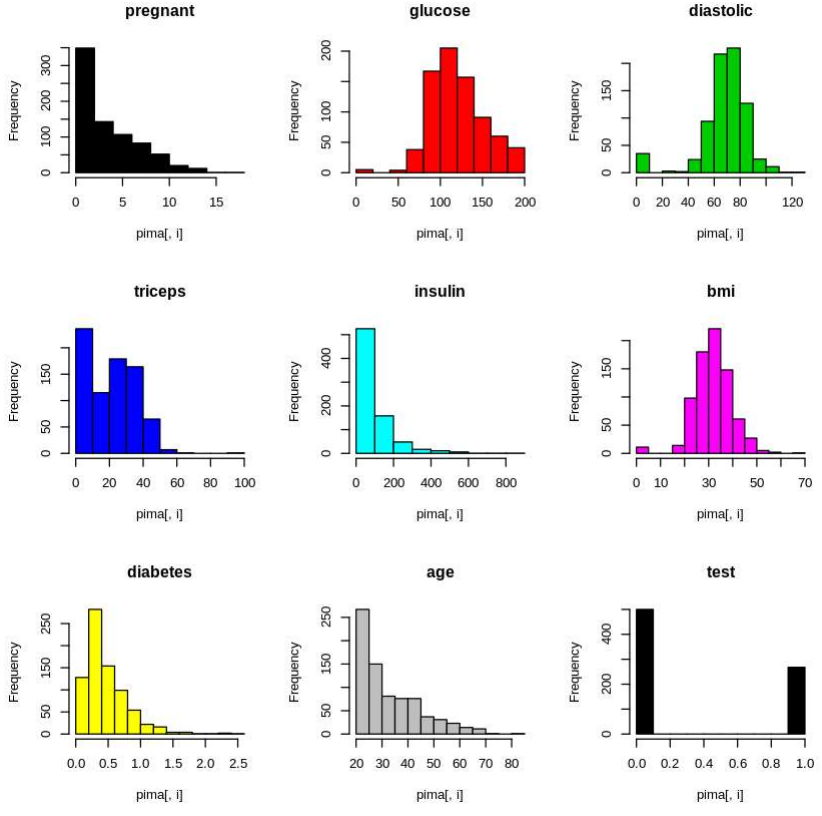
In [3]:

```
# Your Code Here
colSums(is.na(pima))

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

# replace 0s with NAs
# 指定列
metricTraits <- c('glucose', 'diastolic', 'triceps', 'bmi', 'insulin')
pima[metricTraits] <- sapply(pima[metricTraits], function(x) replace(x, x == 0, N
A))
# 移除NA行
pima <- na.omit(pima)

pima <- mutate(pima, test = as.factor(test))

summary(pima)

#
set.seed(1997)

# training set size
n <- floor(0.8 * nrow(pima))

index <- sample(seq_len(nrow(pima)), size = n)
train <- pima[index, ]
test <- pima[-index, ]

# summary(train)
# summary(test)
```

**pregnant:** 0 **glucose:** 0 **diastolic:** 0 **triceps:** 0 **insulin:** 0 **bmi:** 0 **diabetes:** 0 **age:** 0
**test:** 0

| 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 | |




Replaced certain zeros with NAs for data cleaning for measurements ('glucose', 'diastolic', 'triceps', 'bmi', 'insulin') as they shouldn't be 0.

## 1. (b) Initial GLM modelling

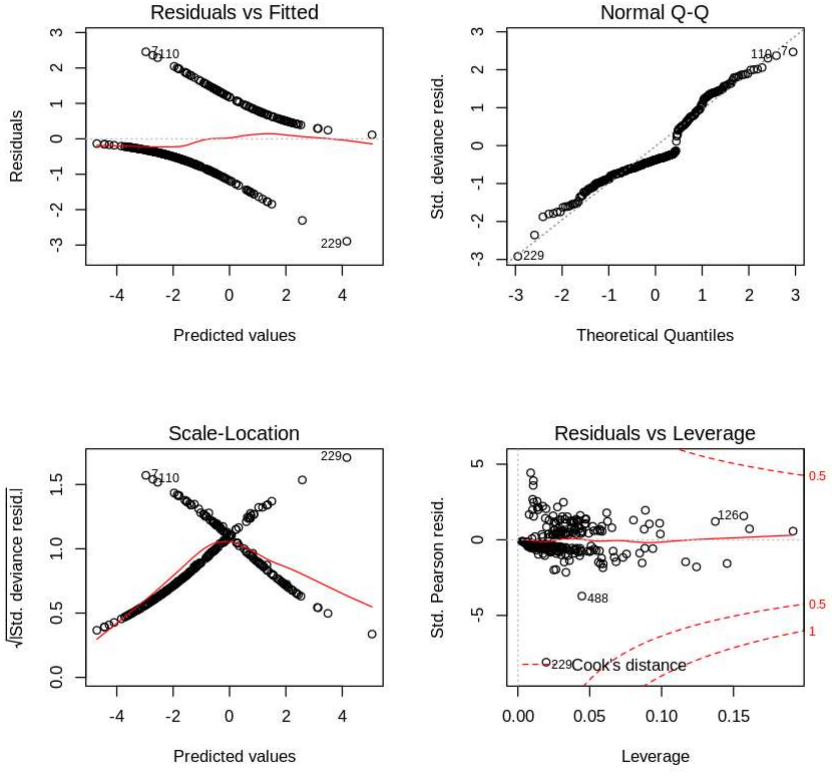
Our data is clean and we're ready to fit! What kind of model should we use to fit these data? Notice that the `test` variable is either $0$ or $1$, for whether the individual tested positive for diabetes. Because `test` is binary, we should use logistic regression (which is a kind of binomial regression).

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 [4]:

```
# Your Code Here
# 创建一个广义线性模型，使用二项分布，以'test'为响应变量，train数据集中的其他变量为解释变量
glmod_pima <- glm(test ~ ., data = train, family = binomial)

summary(glmod_pima)
par(mfrow = c(2,2))
plot(glmod_pima)
```

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

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.8908  -0.6622  -0.3718   0.6218   2.4570

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -9.370e+00  1.326e+00  -7.069 1.56e-12 ***
pregnant     1.141e-01  6.517e-02   1.750   0.0800 .
glucose      3.670e-02  6.646e-03   5.523 3.33e-08 ***
diastolic   -5.983e-03  1.297e-02  -0.461   0.6447
triceps      2.495e-02  1.901e-02   1.312   0.1894
insulin     -9.749e-05  1.485e-03  -0.066   0.9476
bmi          6.098e-02  2.949e-02   2.068   0.0387 *
diabetes     1.059e+00  4.801e-01   2.206   0.0274 *
age          2.127e-02  1.990e-02   1.069   0.2851
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 396.58  on 312  degrees of freedom
Residual deviance: 276.58  on 304  degrees of freedom
AIC: 294.58

Number of Fisher Scoring iterations: 5
```



We cannot test the fit of the model as long as the response is binary, i.e., Y = 0,1 , instead of Y = 0,1, . .n , the residuals will not conform to a normal distribution and the deviations will not conform to a chi-square distribution.We can divide the data into a training set and a test set to see how well the model performs in predicting the values in the test set.

## 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 [5]:
```
# Your Code Here
diastolic_lm = lm(diastolic ~ test, data = train)
summary(diastolic_lm)
```

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

Residuals:
    Min      1Q  Median      3Q     Max
-44.852  -8.852   0.777   8.777  36.777

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   68.852      0.856  80.431  < 2e-16 ***
test1          4.371      1.492   2.929  0.00365 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 12.41 on 311 degrees of freedom
Multiple R-squared:  0.02685,    Adjusted R-squared:  0.02372
F-statistic: 8.579 on 1 and 311 DF,  p-value: 0.003652
```

In the context of the regression model discussed, diastolic blood pressure is not found to be statistically significant. This means that, when other variables are controlled for in the logistic regression model, diastolic blood pressure does not show a significant direct influence on the probability of a positive test result.

On the other hand, descriptive statistics might show that women who test positive tend to have higher diastolic blood pressures on average. This observation suggests a possible association but does not imply a direct causative relationship between higher diastolic pressures and positive test results, as other factors might also be influencing the test outcomes.

The distinction between these two questions lies in the type of conditional probability each addresses:

Is diastolic blood pressure significant in the regression model? This question seeks to understand if there's a statistically significant effect of diastolic blood pressure on the likelihood of testing positive for the condition being studied, independent of other variables. The finding here is that diastolic pressure, by itself, does not significantly alter the odds of a positive test result in the model used.

Do women who test positive have higher diastolic blood pressures? This question addresses whether there is a general pattern or trend where women with positive test results also have higher diastolic pressures. This is more about correlation rather than causation and indicates an observed relationship in the sample data, not controlled for other variables.

The apparent contradiction arises because the first question is answered within the framework of a controlled experimental design (the regression model), which isolates the effect of diastolic blood pressure from other variables. The second question is answered using uncontrolled observational data that do not isolate diastolic blood pressure from other variables. Hence, while there may be a general association observed, it does not translate into a statistically significant effect in a controlled analysis. This discrepancy underscores the importance of understanding the context and methodology behind statistical findings to interpret them correctly.

## 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 [6]:
```
# Your Code Here
summary(glmod_pima)
```

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

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.8908  -0.6622  -0.3718   0.6218   2.4570

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -9.370e+00  1.326e+00  -7.069 1.56e-12 ***
pregnant     1.141e-01  6.517e-02   1.750   0.0800 .
glucose      3.670e-02  6.646e-03   5.523 3.33e-08 ***
diastolic   -5.983e-03  1.297e-02  -0.461   0.6447
triceps      2.495e-02  1.901e-02   1.312   0.1894
insulin     -9.749e-05  1.485e-03  -0.066   0.9476
bmi          6.098e-02  2.949e-02   2.068   0.0387 *
diabetes     1.059e+00  4.801e-01   2.206   0.0274 *
age          2.127e-02  1.990e-02   1.069   0.2851
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 396.58  on 312  degrees of freedom
Residual deviance: 276.58  on 304  degrees of freedom
AIC: 294.58

Number of Fisher Scoring iterations: 5
```

$$\eta = \log\left(\frac{\hat{p}}{1-\hat{p}}\right) = \beta_0 + \beta_1 \text{pregnant} + \beta_2 \text{glucose} + \beta_3 \text{diastolic} + \beta_4 \text{triceps} - \beta_5 \text{insulin} + \beta$$

$$+ \beta_8 \text{age}$$

$$\eta = -10 + 0.1\text{pregnant} + 0.04\text{glucose} - 0.009\text{diastolic} + 0.001\text{triceps} - 0.0003\text{insulin} + 0.0$$

$$+ 1.8\text{diabetes} + 0.04\text{age}$$

When other predictors are taken into account, an increase of one unit in glucose levels corresponds to a 0.04 increase in the log-odds of a positive test. Alternatively, after adjusting for other predictors When adjusting for other predictors, each unit increase in blood glucose level resulted in approximately 1.04

## 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 $2 \times 2$ 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 [7]:
```
# Your Code Here
pr = ifelse(predict.glm(glmod_pima, type = "response", test, na.rm= TRUE) > 0.5,
1, 0)
tn = sum(pr == 0 & as.numeric(levels(test$test))[test$test] == 0);
tp = sum(pr == 1 & as.numeric(levels(test$test))[test$test] == 1);
fp= sum(pr == 1 & as.numeric(levels(test$test))[test$test] == 0);
fn= sum(pr == 0 & as.numeric(levels(test$test))[test$test] == 1);
(tp+tn)/dim(test)[1]

confusion_matrix <- table( Predicted = as.factor(pr), Actual = as.factor(test$tes
t))


print(confusion_matrix)
```

0.759493670886076

```
         Actual
Predicted  0  1
        0 43 10
        1  9 17
```

Here's the confusion matrix

```
         Actual
Predicted  0  1
        0 43 10
        1  9 17
```

## 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 [8]:
```
# Your Code Here
accuracy = (tp+tn)/dim(test)[1]
precision = tp/(tp + fp)
recall = tp/(tp + fn)
F = (2*precision*recall)/(precision + recall)

accuracy
precision
recall
round(F, 4)
```

0.759493670886076

0.653846153846154

0.62962962962963

0.6415

The F-score is a harmonic mean of precision and recall. It's a single metric that combines both precision and recall to give a balanced view of the model's performance. The model's F-score is 64.15%, which is moderate, suggesting a balance between recall and precision but room for improvement.

The model performs moderately well across all metrics, suggesting it has a balanced approach to classifying positives but is not particularly strong in any one area. This might be suitable for general purposes but may require improvement for specific applications where high precision or recall is necessary.

## 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 **Example Scenario Where Accuracy Might Be Misleading** Accuracy might be misleading in scenarios where the dataset is imbalanced between classes. For instance, in a medical diagnostic test for a rare disease where 95% of the cases are negative (no disease) and only 5% are positive, a model that simply predicts "no disease" for all cases will achieve 95% accuracy. However, this model fails to identify any actual cases of the disease, making the high accuracy misleading because the model's performance in detecting positive cases (which is crucial) is extremely poor.

2 **Confusion Matrix for a Response with 3 Levels**

For a classification problem with three categories (say A, B, and C), the confusion matrix would be a 3x3 grid. Each row represents the true classes, while each column represents the predicted classes. The diagonal cells (from top left to bottom right) show the number of correct predictions for each class, and the off-diagonal cells show the misclassifications. Here's how it would look:

| | **Predicted: A** | **Predicted: B** | **Predicted: C** |
|---|---|---|---|
| **True: A** | True Positive A | Misclassified as B | Misclassified as C |
| **True: B** | Misclassified as A | True Positive B | Misclassified as C |
| **True: C** | Misclassified as A | Misclassified as B | True Positive C |

This matrix helps in understanding not just the overall accuracy but also how well the model is performing for each individual class.

3 **Preference Between Type I and Type II Errors in Diabetes Dataset**

In the context of a diabetes dataset, the choice between overestimating Type I error (false positives) and Type II error (false negatives) depends on the implications of each error:

Type I Error (False Positive): Predicting a patient has diabetes when they do not. This might lead to unnecessary stress, further testing, and possibly unwarranted medication, but it ensures that no actual case of diabetes is missed. Type II Error (False Negative): Failing to detect diabetes when the patient actually has the disease. This is more dangerous as it might lead to a lack of necessary treatment and care, potentially resulting in serious health complications. Given these considerations, it would generally be preferable to overestimate Type I error rather than Type II error in a diabetes dataset. This preference is because the consequences of not treating a diabetic patient are typically more severe than the inconvenience and cost of additional tests to confirm a diagnosis where the initial test was a false positive. This approach prioritizes patient safety and the critical need for early intervention in diabetes management.

## 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.

Maya Iskandarani's piece on the ethical challenges surrounding medical consent and privacy, particularly in relation to the Pima Indian Diabetes Data set (PIDD), highlights several key concerns:

1. **Longevity and Scope of Consent**: The PIDD involved data collected from the Pima tribe initially for studying diabetes over an extended period—far beyond the originally intended span. This raises questions about the scope of consent, especially concerning how data intended for one purpose is later used for entirely different research, like refining machine learning algorithms decades later.
2. **Privacy Concerns**: The Pima tribe's data, including sensitive personal information like blood pressure, BMI, and pregnancy counts, has been accessible in the UCI Machine Learning Repository for over two decades. This public availability poses significant privacy risks, particularly when such data can be used in ways participants could not have foreseen at the time of collection.
3. **Eternal Medical Consent**: Iskandarani discusses the concept of "eternal" consent, where participants are asked to agree to the future use of their medical data in ways that are not fully predictable at the time of consent. This concept challenges the traditional understanding of informed consent in medical research, where specifics about the use of data are clearly outlined to participants.
4. **Ethical Controversy and Interdisciplinary Questions**: The availability and use of the Pima data not only provoke ethical debates about consent and privacy but also intertwine with broader discussions across fields such as medical history, anthropology, bioethics, and data analytics. These discussions are crucial in reevaluating how consent is obtained and managed in an era where data can be extensively and perpetually reused and repurposed.

Iskandarani's article serves as a critical reminder of the complexities and ongoing ethical considerations necessary in the management and use of medical data, particularly when it comes to indigenous populations and other vulnerable groups. The piece emphasizes the importance of evolving consent practices to better safeguard privacy and respect the rights and intentions of research participants.

# 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 $\lambda$ is known, is a member of the exponential family.

For an exponential distribution ( Y \sim \text{Exponential}(\lambda) ), the probability density function is given by:

$$f_Y(y) = \lambda e^{-\lambda y} \quad \text{for } y \geq 0$$

We can rewrite this to fit into the exponential family form:

$$f_Y(y) = \lambda e^{-\lambda y} = (1) \cdot \exp(-\lambda y + \log \lambda)$$

Here, we identify:

- $h(y) = 1$ (the base measure),
- $T(y) = y$(the sufficient statistic),
- $\eta(\theta) = -\lambda$ (the natural parameter),
- $A(\theta) = -\log \lambda$ (the log-partition function).

Thus, the exponential distribution fits into the Exponential Family.

## 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.

For a sequence of $n$ independent exponential random variables $Y_i \sim \text{Exponential}(\lambda)$, their sum $Z = \sum_{i=1}^{n} Y_i$ is distributed as a gamma distribution:

$$Z \sim \text{Gamma}(n, \lambda)$$

The pdf of a gamma distribution $Z \sim \text{Gamma}(n, \lambda)$ is given by:

$$f_Z(z) = \frac{\lambda^n}{\Gamma(n)} z^{n-1} e^{-\lambda z} \quad \text{for } z \geq 0$$

We can rewrite this to fit into the exponential family form:

$$f_Z(z) = \frac{\lambda^n}{\Gamma(n)} z^{n-1} e^{-\lambda z} = \frac{\lambda^n}{\Gamma(n)} \exp(-\lambda z + (n-1) \log z)$$

Here, we identify:

- $h(z) = \frac{z^{n-1}}{\Gamma(n)}$ (the base measure),
- $T(z) = z$ (the sufficient statistic),
- $\eta(\theta) = -\lambda$ (the natural parameter),
- $A(\theta) = n \log(\lambda) - \log \Gamma(n)$ (the log-partition function).

Thus, the gamma distribution, and hence the sum of independent exponential random variables, fits into the Exponential Family as well.

These derivations confirm that both the exponential and gamma distributions are part of the Exponential Family, validating their structure and relationship through their pdf forms.

In [ ]: