# 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]:

```r
# 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 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]:
```
# Get a summary of the data
summary(pima)# Your Code Here
```

```
    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  
 Max.   :1.000  
```

In [4]:

```
# Check for zeros in columns where it doesn't make sense
cols_with_zero <- c("glucose", "diastolic", "triceps", "insulin", "bmi")
sapply(pima[cols_with_zero], function(x) sum(x == 0))
```

**glucose:** 5 **diastolic:** 35 **triceps:** 227 **insulin:** 374 **bmi:** 11

In [5]:

```
# Replace zeros with NA in the relevant columns
pima[cols_with_zero] <- lapply(pima[cols_with_zero], function(x) ifelse(x == 0,

# Verify the changes
sapply(pima[cols_with_zero], function(x) sum(is.na(x)))
```

**glucose:** 5 **diastolic:** 35 **triceps:** 227 **insulin:** 374 **bmi:** 11

In [6]:

```
# Impute missing values using the median
library(dplyr)
pima <- pima %>%
  mutate(across(all_of(cols_with_zero), ~ ifelse(is.na(.), median(., na.rm = TRU

# Verify the imputation
sapply(pima[cols_with_zero], function(x) sum(is.na(x)))
```

**glucose:** 0 **diastolic:** 0 **triceps:** 0 **insulin:** 0 **bmi:** 0

In [7]:

```
# Set seed for reproducibility
set.seed(123)

# Split the data
sample_index <- sample(1:nrow(pima), 0.8 * nrow(pima))
train_data <- pima[sample_index, ]
test_data <- pima[-sample_index, ]

# Verify the split
nrow(train_data)
nrow(test_data)
```

614

154

**Explanation of Cleaning Steps**

- **Inspection: We inspected the data to understand its structure and identify potential issues.**
- **Identifying Irregularities: We identified columns where zero values might be nonsensical.**
- **Replacing Zeros with NA: We replaced zero values in specific columns with NA to indicate missing values.**
- **Imputation: We used median imputation to handle missing values, which is a common method to preserve the central tendency without introducing bias.**
- **Splitting Data: We split the data into training and test sets to prepare for model building and evaluation. </strong>**

## 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 [8]:

```
# Fit the logistic regression model
glm_model <- glm(test ~ pregnant + glucose + diastolic + triceps + insulin + bmi
                 data = train_data, family = binomial)

# Summary of the model
summary(glm_model)
```

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

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-2.5345  -0.7206  -0.4056   0.6950   2.3640  

Coefficients:
              Estimate Std. Error z value Pr(>|z|)    
(Intercept) -8.9017449  0.8950353  -9.946  < 2e-16 ***
pregnant     0.1184713  0.0363409   3.260  0.00111 ** 
glucose      0.0389511  0.0044634   8.727  < 2e-16 ***
diastolic   -0.0120342  0.0096359  -1.249  0.21170    
triceps      0.0001204  0.0144736   0.008  0.99336    
insulin     -0.0010706  0.0012503  -0.856  0.39183    
bmi          0.0945454  0.0200427   4.717 2.39e-06 ***
diabetes     0.7121157  0.3205776   2.221  0.02633 *  
age          0.0139045  0.0106135   1.310  0.19017    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 796.42  on 613  degrees of freedom
Residual deviance: 570.09  on 605  degrees of freedom
AIC: 588.09

Number of Fisher Scoring iterations: 5
```

**If the model summary shows significant predictors (e.g., glucose, bmi, diabetes) with p-values less than 0.05 and a substantial reduction in deviance, the model can be considered to fit 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 [9]:

```
# Check the summary of the logistic regression model
summary(glm_model)
```

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

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-2.5345  -0.7206  -0.4056   0.6950   2.3640  

Coefficients:
              Estimate Std. Error z value Pr(>|z|)    
(Intercept) -8.9017449  0.8950353  -9.946  < 2e-16 ***
pregnant     0.1184713  0.0363409   3.260  0.00111 ** 
glucose      0.0389511  0.0044634   8.727  < 2e-16 ***
diastolic   -0.0120342  0.0096359  -1.249  0.21170    
triceps      0.0001204  0.0144736   0.008  0.99336    
insulin     -0.0010706  0.0012503  -0.856  0.39183    
bmi          0.0945454  0.0200427   4.717 2.39e-06 ***
diabetes     0.7121157  0.3205776   2.221  0.02633 *  
age          0.0139045  0.0106135   1.310  0.19017    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 796.42  on 613  degrees of freedom
Residual deviance: 570.09  on 605  degrees of freedom
AIC: 588.09

Number of Fisher Scoring iterations: 5
```

**Question 1: Is diastolic blood pressure significant in the regression model?**

**To determine the significance of diastolic blood pressure in the regression model, refer to the p-value associated with the diastolic blood pressure coefficient in the logistic regression model summary. It is considered NOT statistically significant.**

**Question 2: Do women who test positive have higher diastolic blood pressures?**

**To address whether women who test positive have higher diastolic blood pressures, we can compare the average diastolic blood pressure for women who tested positive (test = 1) against those who tested negative (test = 0).**

In [10]:

```
# Compare diastolic blood pressure between positive and negative test groups
mean_positive <- mean(train_data$diastolic[train_data$test == 1], na.rm = TRUE)
mean_negative <- mean(train_data$diastolic[train_data$test == 0], na.rm = TRUE)
mean_positive
mean_negative
```

74.4398148148148

70.5954773869347

**Distinction and Discussion**

**The two questions address different aspects:**

1. **Significance in the Model: This examines whether diastolic blood pressure is a statistically significant predictor of diabetes in the context of the logistic regression model, considering all other predictors. This is a conditional analysis based on the presence of other variables in the model.**

2. **Comparative Averages: This looks at the average diastolic blood pressure in isolation between the two groups (positive and negative test results). This is a marginal analysis.**

## 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.

$$\log\left(\frac{P(\text{test}=1)}{1-P(\text{test}=1)}\right) = \beta_0 + \beta_1 \cdot \text{pregnant} + \beta_2 \cdot \text{glucose} + \beta_3 \cdot \text{diastolic} + \beta_4 \cdot \text{trice}$$

**Where:**

$\beta_0$ **is the intercept.**

$\beta_1, \beta_2, \ldots., \beta_8$ **are the coefficients for the predictors.**

**In words, a one-unit increase in glucose (assuming all other predictors are held constant) results in a change in the log-odds of testing positive for diabetes by the amount of the coefficient of glucose. Specifically, if the coefficient of glucose ($\beta_2$) is positive, it indicates that higher glucose levels are associated with a higher probability of testing positive for diabetes.**

## 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 [11]:

```r
# Predict the probabilities on the test set
test_pred_prob <- predict(glm_model, newdata = test_data, type = "response")

# Convert probabilities to binary outcomes using 0.5 as the threshold
test_pred <- ifelse(test_pred_prob > 0.5, 1, 0)
```

In [12]:

```r
# Create the confusion matrix
confusion_matrix <- table(Predicted = test_pred, Actual = test_data$test)

# Print the confusion matrix
confusion_matrix
```

```
          Actual
Predicted  0  1
        0 90 23
        1 12 29
```

## 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 [13]:
# Calculate accuracy, precision, recall, and F1-score
TP <- confusion_matrix[2, 2]
TN <- confusion_matrix[1, 1]
FP <- confusion_matrix[2, 1]
FN <- confusion_matrix[1, 2]

accuracy <- (TP + TN) / sum(confusion_matrix)
precision <- TP / (TP + FP)
recall <- TP / (TP + FN)
f1_score <- 2 * (precision * recall) / (precision + recall)

# Print the metrics
list(accuracy = accuracy, precision = precision, recall = recall, f1_score = f1_
```

```
$accuracy     0.772727272727273
$precision    0.707317073170732
$recall       0.557692307692308
$f1_score     0.623655913978495
```

**The model shows a reasonable fit with an accuracy of about 77.27%. However, the recall value is relatively low, indicating that the model might be missing a considerable number of actual positive cases. This suggests a need for further refinement of the model, such as feature selection, tuning hyperparameters, or using more advanced modeling techniques to improve recall without compromising precision too much.**

## 1. (g) Understanding Evaluation Statistics

Answer the following questions in the markdown cell below.

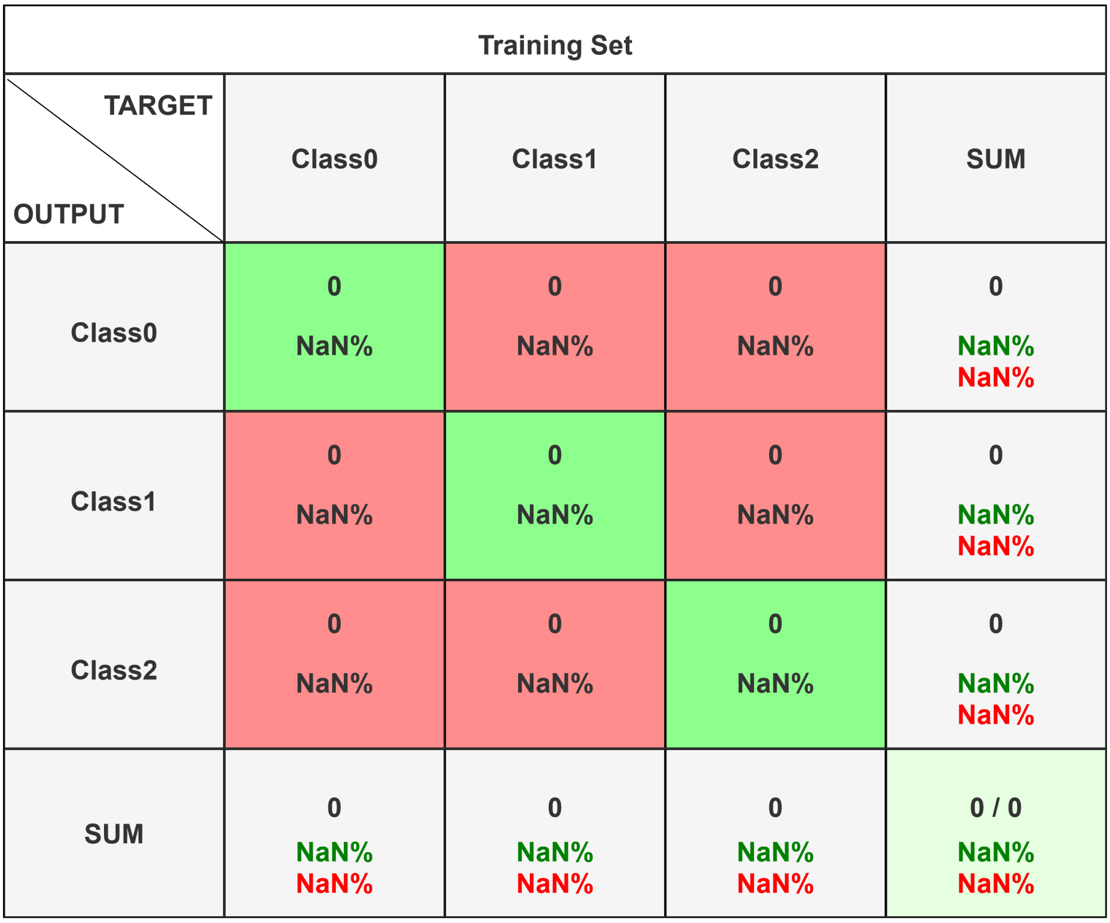
1. Give an example scenario for when accuracy would be a misleading evaulation statistic.
2. Confusion matrices can also be used for non-binary classification problems. Describe what a confusion matrix would look like for a response with $3$ levels.
3. You'll have to take our word on the fact (or spend 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 for Misleading Accuracy

Accuracy can be misleading in cases where the dataset is imbalanced, meaning one class is much more frequent than the other. For example, consider a medical diagnostic test for a rare disease that occurs in only 1% of the population. If the model simply predicts that no one has the disease, it will be 99% accurate because it correctly identifies the 99% who do not have the disease. However, it fails to identify any actual cases of the disease, which is a critical failure for a medical test.

1. Confusion Matrix for a Response with 3 Levels

For a response with three levels (e.g., Class 0, Class 1, Class 2), the confusion matrix would be a 3x3 matrix:



| Training Set | | | | |
|---|---|---|---|---|
| TARGET<br>OUTPUT | Class0 | Class1 | Class2 | SUM |
| Class0 | 0<br>NaN% | 0<br>NaN% | 0<br>NaN% | 0<br>NaN%<br>NaN% |
| Class1 | 0<br>NaN% | 0<br>NaN% | 0<br>NaN% | 0<br>NaN%<br>NaN% |
| Class2 | 0<br>NaN% | 0<br>NaN% | 0<br>NaN% | 0<br>NaN%<br>NaN% |
| SUM | 0<br>NaN%<br>NaN% | 0<br>NaN%<br>NaN% | 0<br>NaN%<br>NaN% | 0 / 0<br>NaN%<br>NaN% |

Where:

- TP_0, TP_1, TP_2: True Positives for Class 0, 1, and 2 respectively.
- FP_1_0: False Positives where Class 1 is predicted as Class 0.
- FP_0_1: False Positives where Class 0 is predicted as Class 1.
- FP_2_0: False Positives where Class 2 is predicted as Class 0.
- And similarly for other entries.

1. Preference Between Type I and Type II Error in Diabetes Dataset

In the context of the diabetes dataset, preferring a model that overestimates Type I error (false positives) or Type II error (false negatives) depends on the

**consequences of these errors:**

- **Type I Error (False Positive): The model predicts diabetes when the person does not have it. This could lead to unnecessary anxiety and medical tests for the individual.**
- **Type II Error (False Negative): The model fails to predict diabetes when the person actually has it. This could lead to missed diagnoses and lack of necessary treatment, which can have serious health consequences.**

**Given the severe health implications of untreated diabetes, it is generally more critical to avoid Type II errors. Therefore, it would be preferable to have a model that overestimates Type I error (false positives) rather than Type II error (false negatives). Ensuring that individuals who might have diabetes are correctly identified for further testing and potential treatment is more important than the inconvenience of false positives.**

## 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.

**Maya Iskandarani's piece on consent and privacy concerns raised several issues regarding the Pima Indian Diabetes Dataset (PIDD):**

1. **Informed Consent: The Pima tribe participated in a long-term observational study initiated by the NIH, originally meant to last 10 years but extended to 40 years. The complexity and duration of such studies make it challenging to provide participants with comprehensive information about how their data will be used in the future. This raises concerns about the adequacy of informed consent.**

2. **Data Privacy: The PIDD has been publicly accessible for over two decades, containing sensitive personal information such as blood pressure, BMI, and number of pregnancies. While the dataset is valuable for refining machine learning algorithms to predict and prevent diabetes, the public availability of such detailed personal data poses significant privacy risks.**

3. **Ethical Controversy: The long-term use and public accessibility of the PIDD highlight the ethical controversy around using historical and personal data without ongoing consent from the participants. Researchers cannot realistically inform participants about all future uses of their data, leading to concerns about "eternal" medical consent.**

4. **Interdisciplinary Questions: The case of the Pima tribe illustrates broader interdisciplinary questions at the intersection of medical history, anthropology, bioethics, and data analytics. It challenges researchers to consider the long-term ethical implications of data collection and usage beyond immediate scientific goals.**

**These concerns underscore the importance of establishing robust ethical guidelines and consent processes for the collection and use of personal data in medical research, particularly when such data is used for long-term studies and made publicly accessible.**

# 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.

To show that $(Y \sim \text{exponential}(\lambda)$ belongs to the exponential family of distributions, we start with its probability density function (pdf):

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

The general form for a distribution in the exponential family is:

$f(y; \theta) = h(y) \exp\left(\frac{y\theta - c(\theta)}{\phi}\right)$

Let's rewrite $(f(y; \lambda)$ in this form:

1. **Identify the components:**

   - $(\theta = \lambda$
   - $(h(y) = 1), becausethere'snoadditionalfunctionof\backslash(y\backslash)intheexponentialte$
   - $(c(\theta) = \frac{1}{\lambda}), derivedfromthemomentgeneratingfunction(MGF)oftheexpon$

1. **Rewrite the pdf:**

   The pdf $f(y; \lambda) = \lambda e^{-\lambda y}$ can be expressed as:

   $f(y; \lambda) = 1 \cdot e^{-\lambda y} \cdot \lambda$

   Here,

   - $(h(y) = 1$
   - $\frac{y\theta - c(\theta)}{\phi} = -\lambda y$

   where $\theta = \lambda), (c(\lambda) = \frac{1}{\lambda}), and \phi = 1.$

Therefore, $Y \sim \text{exponential}(\lambda)$ can indeed be expressed in the form that characterizes the exponential family of distributions:

$f(y; \lambda) = \lambda e^{-\lambda y} = \exp\left(\frac{y\lambda - \frac{1}{\lambda}}{1}\right) \cdot 1$

Thus, $Y \sim \text{exponential}(\lambda)$ belongs to the exponential family of distributions.

## 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.

To show that $Z = \sum_{i=1}^{n} Y_i$ follows the form of the exponential family distribution, let's express it in a form that aligns with the general definition of the exponential family.

Given:

- $Y_i \sim \text{Exponential}(\lambda)$, with probability density function $f_{Y_i}(y_i; \lambda) = \lambda e^{-\lambda y_i}, \quad y_i \geq 0$.
- $Z = \sum_{i=1}^{n} Y_i$.

First, the joint distribution of $Y_1, \ldots, Y_n$:

$f_{Y_1,\ldots,Y_n}(y_1, \ldots, y_n; \lambda) = \prod_{i=1}^{n} \lambda e^{-\lambda y_i} = \lambda^n e^{-\lambda \sum_{i=1}^{n} y_i}$.

Now, the distribution of ( Z ):

$f_Z(z; \lambda) = \int_{y_1+\cdots+y_n=z} \lambda^n e^{-\lambda z} \, dy_1 \cdots dy_n$.

To express $f_Z(z; \lambda)$ in the form of an exponential family distribution, consider: $f_Z(z; \lambda) = \lambda^n e^{-\lambda z} \cdot \mathbf{1}\{z \geq 0\}$.

This can be written as: $f_Z(z; \lambda) = \exp(n \log \lambda - \lambda z) \cdot \mathbf{1}\{z \geq 0\}$.

Therefore, $Z \sim \text{Gamma}(n, \lambda)$ can be expressed in the exponential family form: $f_Z(z; \lambda) = \exp\left(\frac{z \cdot \theta - b(\theta)}{a(\phi)} + c(z, \phi)\right)$,

where:

- $\theta = \lambda$,
- $b(\theta) = n \log \lambda$,
- $a(\phi) = 1$,
- $c(z, \phi) = 0$.

Thus, $Z$ belongs to the exponential family with parameters $\theta = \lambda, \phi = 1$, sufficient statistic $T(z) = z$, natural parameter $\eta = \log \lambda$, and log-partition function $A(\eta) = -n \log \lambda$.

Therefore, we have shown that $Z = \sum_{i=1}^{n} Y_i \sim \text{Gamma}(n, \lambda)$ is indeed a member of the exponential family.